

Sistema de Gestão e Manutenção Aplicado à Indústria Cerâmica – Estágio na SA - Soluções em Automação S.A.

Relatório de estágio apresentado para a obtenção do grau de Mestre em
Engenharia Eletrotécnica - Área de Especialização em Automação e
Comunicações em Sistemas Industriais

Autor

Carlos Filipe de Oliveira Saraiva

Orientadores

Frederico Miguel do Céu Marques dos Santos

João Paulo Morais Ferreira

Tutor

André Henriques Simões Seabra da Costa

Instituto Superior de Engenharia de Coimbra

Coimbra, junho 2015

AGRADECIMENTOS

Este relatório de estágio, resultou de um trajeto longo, com momentos complicados, mas ao longo do qual fui recebendo o maior apoio e estímulo de muitos. Neste sentido, os méritos que possa haver, devem-se aos contributos de várias pessoas que me ajudaram de diversas formas.

Em primeiro lugar agradecer aos meus pais Carlos Melo Saraiva e Maria Edite Saraiva, pelo esforço de alguns anos e por todo o apoio que sempre me deram em todas as ocasiões. Agradecer ao meu irmão Bruno Saraiva, à minha cunhada Susana Figueiredo e aos meus grandes amigos Rafael Pereira, Paulo Melo e Raquel Tomé.

Agradeço aos meus orientadores Prof. Doutor João Paulo Morais Ferreira e Prof. Doutor Frederico Miguel do Céu Marques dos Santos pela ajuda e pelo tempo a que os obriguei, em parte involuntariamente, a esperar por este relatório.

O meu reconhecimento vai também para todos os colaboradores da empresa SA – Soluções em Automação S.A., em especial ao Eng. Pedro Pedrosa e Eng. André Costa por todo o apoio durante a realização deste projeto.

Por fim quero agradecer a todos os meus colegas de Mestrado, por todos os momentos partilhados e companheirismo existente. Em especial tenho de agradecer ao meu amigo André Correia, pois ele faz parte deste meu trajeto desde o início. Agradecer também aos meus amigos Francisco Rodrigues, Elca Gonçalves e Claudia Alves por todos os trabalhos realizados em grupo, todo o apoio que sempre me deram que não poderia esquecer

RESUMO

Este trabalho tem como objetivo o desenvolvimento de um Sistema de Gestão de Manutenção para a indústria cerâmica, através de um estágio curricular realizado na empresa SA – Soluções em Automação S.A., que permita a gestão eficiente de ordens de trabalho de manutenção de uma empresa, sejam elas do tipo preventiva, de melhoria ou corretiva, bem como a gestão da documentação inerente.

O setor cerâmico é um setor eminentemente exportador possuindo um longo historial de negócios internacionais, sendo natural que ambição de melhoria se situe na equivalência às melhores práticas e soluções, pelo que este sistema de gestão e manutenção se encaixa totalmente nesta ideia.

Foi realizada uma revisão bibliográfica, relativa à gestão da manutenção e ao *software* utilizado para melhor preparar o desenvolvimento do projeto, apresentando as vantagens da sua utilização, assim como uma comparação relativamente a sistemas similares.

O trabalho foi realizado no âmbito no estágio curricular do mestrado em Engenharia Eletrotécnica - Área de Especialização em Automação e Comunicações em Sistemas Industriais na empresa SA – Soluções em Automação, S.A.

Com a experiência no ramo por parte dos colaboradores da empresa, existiu um levantamento das necessidades e uma familiarização com o caso em estudo na indústria cerâmica, de forma a garantir a melhor solução a implementar. As necessidades em causa foram analisadas e com base nas mesmas, procurou-se aplicar alguns conceitos que possibilitam organizar e dinamizar o sistema. Estes conceitos têm como base, toda a estrutura, funcionamento e processos utilizados numa fábrica deste tipo de indústria, bem como a maquinaria principal, como fornos, secadores e seus sub-equipamentos.

Feito o levantamento e respetiva análise de requisitos, procedeu-se a conceção do sistema informático de gestão e manutenção, usando uma estratégia geral e abrangente, assente em processos de alto nível, modelos de dados e *interface* com o utilizador. Construído o *interface* com o utilizador, encontra-se dinamizado o sistema, permitindo a informatização dos processos de gestão da manutenção.

Em termos práticos, foi criado um mecanismo de gestão de processos e manutenção de equipamentos, gestão de equipas, material, *schedules* e ordens de manutenção, tudo isto visando a indústria cerâmica. Adicionalmente foi criado também um sistema de relatórios para melhorar a análise e a respetiva gestão dos processos.

O Sistema de Gestão de Manutenção foi colocado em funcionamento num Cliente da empresa SA – Soluções em Automação, S.A. sendo realizadas também as respetivas análises e conclusões ao trabalho efetuado, provando ser uma sistema fundamental para uma boa estrutura, organização e bom funcionamento de qualquer fábrica de indústria cerâmica.

Palavras-Chave

Manutenção, Gestão, Sistema, Equipamentos, Ordens de Trabalho, Indústria, Cerâmica, Dinamizar, Processo

ABSTRACT

The purpose of this project is the development of a Management and Maintenance System for the ceramic industry, through a curricular training held at the company SA - Soluções em Automação S.A., allowing an efficient supervision of maintenance work orders in a company, whether they are of preventive, improvement or corrective type, as well as the administration of inherent documentation.

The ceramic sector is predominantly an export sector having a long history of international business, it is natural that improvement ambition lies in equivalence to the best practices and solutions, so this management and maintenance system fits completely in this idea.

A literature review was conducted regarding the maintenance management and the software used to better prepare the development of the project, presenting the advantages of its use, as well as a comparison for similar systems.

The work was performed under the curricular internship of the Electronic Engineering masters - Specialization in Automation and Communications in Industrial Systems, in the company SA-Soluções em Automação, S.A.

With the experience of the employees in the business, it was made an assessment of the needs and a familiarization with the case study in the ceramic industry, in order to ensure the best solution to implement. The detected flaws were analyzed and based on them we attempted to apply some concepts that allow organizing and boosting the System. These concepts are based on the entire structure, operation and procedures used in a factory of this type of industry as well as the main equipment, such as ovens, dryers and its sub-equipments.

Then, after the assessment and respective analysis, we decided to design the computer System of management and maintenance, using a general and comprehensive strategy, based on high-level processes, data models and interface with the user. Built the interface with the user, the System is boosted, allowing the computerization of the maintenance management procedures.

In practical terms it was created a mechanism for device maintenance procedures management, teams' management, schedules and maintenance orders, all this aiming the ceramic industry. Additionally, a reporting System was also created to improve the analysis and respective processes management.

The Maintenance Management System was put to work in a SA - Soluções em Automação, S.A. company's Customer, being carried out the respective analyses and conclusions to the performed work, proving to be a fundamental system for a good structure, organization and proper functioning of any ceramic industry factory.

Keywords

Maintenance, Management, *System*, Devices, Work Orders, Industry, Ceramic, To Boost, Process

ÍNDICE

CAPITULO 1 - INTRODUÇÃO	1
1.1. Motivação	1
1.2. Objetivos	2
1.3. Organização do Trabalho	3
 CAPITULO 2 - REVISÃO DA LITERATURA	 5
 CAPITULO 3 - SOLUÇÃO WONDERWARE	 9
3.1. <i>Wonderware InTouch</i>	9
3.2. <i>Wonderware System Platform</i>	11
3.3. <i>Wonderware Historian</i>	14
 CAPITULO 4 - SISTEMA DE GESTÃO E MANUTENÇÃO DESENVOLVIDO	 19
4.1. <i>Microsoft SQL Server</i>	20
4.1.1 Tabelas SQL	22
4.1.2 Views SQL	28
4.1.3 <i>Stored Procedures SQL</i>	30
4.2. <i>System Platform</i>	35
4.3. <i>InTouch HMI</i>	58
 CAPITULO 5 - RESULTADOS E CONCLUSÕES	 69
5.1 Resultados	69
5.2 Conclusões	70
 REFERÊNCIAS BIBLIOGRÁFICAS	 73
 ANEXOS -TABELAS MICROSOFT SQL SERVER	 75

ÍNDICE DE FIGURAS

Figura 1.1: Logótipo da Empresa SA - Soluções em Automação, S.A.....	1
Figura 1.2: Diagrama demonstrando o fluxo de trabalho [23] associado à gestão e manutenção de ativos.....	2
Figura 3.1: Exemplo de uma aplicação realizada em Wonderware InTouch.....	10
Figura 3.2: Diagrama funcional do System Platform relativamente aos softwares da Wonderware.....	12
Figura 3.3: Diagrama representando a interatividade do Historian [8].....	15
Figura 3.4: Ferramenta Trend do Historian Client [8].....	16
Figura 4.1: Diagrama funcional com as interligações entre os softwares utilizados.....	19
Figura 4.2: Diagrama da base de dados gestão de manutenção desenvolvida.	21
Figura 4.3: Tabelas SQL Server de Equipamento, Conjunto e SubConjunto.	22
Figura 4.4: Tabela de Peças em SQL Server.	23
Figura 4.5: Tabela Schedule da base de dados de gestão e manutenção.	23
Figura 4.6: Tabelas referentes aos Estados da <i>Schedule</i> , Tipo de <i>Schedule</i> e Tipo de Execução em SQL Server.	24
Figura 4.7: Tabela de Ordens de Manutenção do sistema de gestão e manutenção.....	25
Figura 4.8: Tabelas de material para as schedules e ordens de manutenção.	26
Figura 4.9: Tabelas ScheduleActividade e Actividade da base de dados.....	27
Figura 4.10: Tabelas onde é guardada toda a documentação inerente ao sistema.....	28
Figura 4.11: View SQL apresentando todas as schedules em agenda.....	29
Figura 4.12: Procedimento Modificar na tabela de Equipamentos.	31
Figura 4.13: Stored Procedure utilizado para fechar uma ordem de manutenção.	34
Figura 4.14: Ferramenta de design Archestra IDE.....	35
Figura 4.15: Script de tradução da aplicação usado no objeto Manutenção.....	37
Figura 4.16: Script .NET representando o procedimento Adicionar.	38
Figura 4.17: Janela de Equipamentos do objeto com o mesmo nome.....	39
Figura 4.18: Janela de popup de introdução de dados do objeto Equipamentos.	40
Figura 4.19: Janela de Documentação do objeto Equipamentos.	40
Figura 4.20: Símbolo gráfico principal do objeto Peças.	42
Figura 4.21: Scripts utilizados no símbolo gráfico de popup de Peças.	43
Figura 4.22: Script utilizado para mostrar os dados no datagrid relativos às schedules.	44
Figura 4.23: Símbolo de popup de introdução de dados das schedules.	45
Figura 4.24: Símbolo de material relativo aos agendamentos.....	46
Figura 4.25: Popup final de operações relativas com schedules.	47
Figura 4.26: Janela de documentação do objeto Schedule.	48
Figura 4.27: Janela principal do objeto Ordens de Manutenção.	49
Figura 4.28: Símbolo gráfico relativo a uma ordem de manutenção gerada de uma schedule.....	50
Figura 4.29: Símbolo gráfico relativo a ordens de manutenção geradas a partir de schedules.....	51

Figura 4.30: Símbolo gráfico referentes aos documentos relacionados com ordens de manutenção.....	53
Figura 4.31: Construção do relatório relativo à listagem de equipamentos existente.	54
Figura 4.32: Relatório de tarefas agendadas por equipamento.....	55
Figura 4.33: Relatório relativo a uma schedule individual com toda a informação associada.	56
Figura 4.34: Relatório apresentando todas as ordens de manutenção e os seus estados relativamente às datas escolhidas.	57
Figura 4.35: Menu relativo ao sistema de gestão e manutenção da aplicação	58
Figura 4.36: Ambiente gráfico InTouch com as janelas da aplicação de gestão e manutenção definidas.	59
Figura 4.37: Janela de Equipamentos relativa à supervisão em InTouch.....	59
Figura 4.38: Janela de popup de modificação de dados dos equipamentos.	60
Figura 4.39: Janela de Documentos relativo aos equipamentos.....	60
Figura 4.40: Janela de Peças da aplicação de gestão e manutenção.....	61
Figura 4.41: Popup de adição, modificação ou remoção de material.....	61
Figura 4.42: Janela de Equipas na aplicação de gestão e manutenção.	62
Figura 4.43: Janela de Schedules da supervisão de gestão e manutenção.....	63
Figura 4.44: Popup relativo aos dados de uma tarefa em agenda.....	64
Figura 4.45: Popup de documentação inerente á schedule.....	64
Figura 4.46: Janela de ordens de manutenção da supervisão.	65
Figura 4.47: Janela de documentação relacionada com as ordens de manutenção.	67
Figura 4.48: Janela de relatórios do sistema.....	67

ABREVIATURAS

API	<i>Application Programming Interface</i>
BD	Base de Dados
CAPEX	<i>Capital Expenditure</i>
DCS	<i>Distributed Control System</i>
HMI	<i>Human-Machine Interface</i>
I/O	<i>Input / Output</i>
IDE	<i>Integrated Development Environment</i>
IED	<i>Improvised Explosive Device</i>
IT	<i>Information Technology</i>
OLE	<i>Object Linking and Embedding</i>
OM	Ordem de Manutenção
OPC	<i>OLE for Process Control</i>
OT	Ordem de Trabalho
PLC	<i>Programmable Logic Controller</i>
ROI	<i>Return on Investment</i>
RTU	<i>Remote Terminal Unit</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SQL	<i>Structured Query Language</i>
UDA	<i>User Defined Object</i>

1. INTRODUÇÃO

Neste relatório de estágio pretende-se desenvolver uma plataforma de gestão e manutenção para uma cerâmica baseado em *software* SCADA *Wonderware*, capaz de melhorar as competências e práticas de gestão do setor melhorando a eficiência e produção da empresa.

1.1. Motivação

Nos últimos anos tem-se assistido a uma enorme proliferação no que diz respeito a ferramentas e *softwares* de gestão e manutenção. O uso destas ferramentas tornou-se muitas vezes generalizado cabendo ao utilizador adaptar ao seu caso em particular. Sabendo que o sector cerâmico é o principal foco da empresa SA - Soluções em Automação, S.A. [22] (Figura 1.1) e que todas as ferramentas de supervisão são elaboradas e construídas recorrendo ao *software* *Supervisory Control and Data Acquisition* (SCADA) da *Wonderware*, pensou-se em projetar um sistema de gestão e manutenção totalmente adaptado às necessidades deste tipo de Cliente.

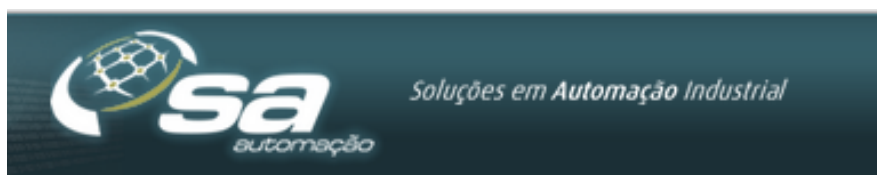


Figura 1.1: Logótipo da Empresa SA - Soluções em Automação, S.A.

Sabe-se que sendo o setor cerâmico um setor eminentemente exportador possuindo um longo historial de negócios internacionais, é natural que a sua ambição de melhoria se situe na equivalência às melhores práticas que se desenvolvem nos mercados mais importantes. Esta ideia torna-se bastante importante quando se pensa nos melhoramentos essenciais por parte da empresa de forma a tornar essa ambição legítima, possibilitando aos Clientes a melhor solução possível.

Com uma solução de gestão e manutenção é permitido ao Cliente gerir e simplificar todo o tipo de ações que ocorrem no dia-a-dia com os recursos e equipamentos da empresa, podendo atuar ao nível de manutenções preventivas, corretivas e correntes. Este facto trará uma maior eficiência à empresa ajudando num melhor controlo e oferecendo aos encarregados da manutenção a possibilidade de programarem as tarefas antecipadamente baseando-se em períodos, datas ou contadores existentes.

A integração de todo este sistema de gestão e manutenção numa aplicação de supervisão SCADA *Wonderware*, torna este processo mais ambicioso pois contrasta de toda a gama de sistemas de gestão de manutenção e ativos existentes no mercado e oferece igualmente uma solução completa para a gestão eficiente da empresa.

1.2. Objetivos

Para este estágio foi proposto o desenvolvimento de um sistema capaz de dar resposta às necessidades correntes de gestão e manutenção de uma empresa cerâmica. Apesar deste objetivo principal, como todo o sistema será uma aplicação de supervisão concebida com recurso ao *software* SCADA *Wonderware*, outro dos objetivos passa por criar um ambiente gráfico apelativo, intuitivo e simples de trabalhar recorrendo a todas as ferramentas que o *software* possibilita.

Este sistema tem de ser capaz de implementar os vários mecanismos relacionados com a gestão da manutenção, gestão de recursos e alocação de equipamentos [14]. A gestão de equipamentos é fundamental para operações que têm como objetivo, alcançar as melhores decisões de rentabilidade e rapidez na alocação de recursos. O sistema tem de ser capaz também de prevenir tempos de paragem, podendo gerar automaticamente as ordens de serviço de manutenção preventiva para organizar efetivamente o fluxo de trabalho, considerando o tempo de paragem, além de controlar a finalização das tarefas. Na Figura 1.2 é mostrado um diagrama que pretende traduzir o fluxo de trabalho inerente à manutenção de ativos. O objetivo é programar estrategicamente as tarefas preventivas e preditivas dentro de um tempo produtivo de maneira a maximizar o uso dos equipamentos e minimizar esse mesmo tempo de paragem, além de organizar eficientemente a mão-de-obra para evitar o pagamento de horas extras.



Figura 1.2: Diagrama demonstrando o fluxo de trabalho associado à gestão e manutenção de ativos [23].

O custo é o fator que a maioria dos altos executivos associa à manutenção de equipamentos e ativos. Infelizmente, muitas empresas ainda não estão absorveram a ideia de que a manutenção tem um grande potencial para o aumento de produtividade e lucratividade. A manutenção, embora seja muitas vezes retirada da lista de prioridades de investimentos, apresenta um potencial de retorno do investimento, extremamente atraente.

Como alguns dos potenciais benefícios para o Cliente deste sistema de gestão e manutenção proposto, encontra-se o aumento da capacidade produtiva, aumento da vida útil dos equipamentos e ativos, aumento da eficiência das tarefas de manutenção, redução do custo operacional e a disponibilização de uma informação mais precisa e completa para uma melhor tomada de decisão.

Como objetivo final encontra-se o comissionamento e validação em obra da aplicação de gestão e manutenção desenvolvida e análise dos respectivos resultados e testes efetuados.

1.3. Organização do Trabalho

Este trabalho encontra-se organizado em cinco capítulos, sendo que o primeiro é focado na motivação e contextualização do problema, ou seja, explicação da necessidade que levou ao seu desenvolvimento, análise dos procedimentos de trabalho existentes e naquilo que deve ser a organização e os conceitos a aplicar na gestão de ordens de trabalho de manutenção, bem como os objetivos propostos para o projeto.

No segundo capítulo é realizada uma revisão bibliográfica, assente em conceitos de manutenção e gestão, apresentando a solução SCADA utilizada, no caso, solução *Wonderware*, bem como as suas vantagens como solução independente relativamente a concorrentes, quais as suas maiores utilizações nos diversos mercados mundiais e o porquê de se apresentar como uma solução SCADA/HMI bastante confiável.

O terceiro capítulo é relativo à apresentação os *softwares* e soluções *Wonderware* estudadas, a sua importância para os vários tipos de indústria existentes, principais recursos e características técnicas que fazem destas ferramentas as mais utilizadas a nível mundial, bem como as suas grandes vantagens e benefícios para a empresa e utilizadores.

No quarto capítulo surge de uma forma geral a apresentação da aplicação desenvolvida para a gestão e manutenção de uma empresa da indústria cerâmica. Neste capítulo será mostrado todo o processo de construção da aplicação desde o seu início até a fase de validação, as várias relações e dependências existentes e a modelação e detalhes de implementação da solução desenvolvimento.

No quinto capítulo surgem os resultados, avaliações e testes efetuados aquando do comissionamento e validação da aplicação desenvolvida, validação esta que será realizada remotamente. Serão apresentadas também as conclusões ao trabalho realizado, bem como sugestões para desenvolvimento e trabalho futuro.

2. REVISÃO DA LITERATURA

Historicamente, nos meados das décadas de 70 e 80, com o avanço da tecnologia e desenvolvimento dos primeiros microprocessadores, o computador tornou-se cada vez mais numa peça chave nos diversos setores industriais, começando as empresas a utilizá-los no planeamento das atividades de manutenção, bem como, no controlo do inventário, recolha e armazenamento de dados, históricos de processos e apoio ao sistema de custos industriais [26].

No princípio dos anos 90 foi introduzida a abordagem da manutenção baseada em fiabilidade, que combina de uma forma eficaz as forças de todas estas estratégias e filosofias num único sistema de manutenção [24].

O último século assistiu assim a uma enorme evolução na sofisticação das máquinas utilizadas nos processos produtivos, provocada principalmente pela obrigatoriedade de aumento de produtividade, levando a que os equipamentos evoluíssem de sistemas puramente mecânicos para sistemas eletromecânicos com sofisticados controlos por computador, chamados sistemas de supervisão.

As empresas estão cada vez mais, sujeitas a um ambiente altamente competitivo, sendo que o fator de diferenciação passa pelo desempenho operacional, que se baseia na melhoria contínua dos processos, sendo que este ambiente faz com que estas mesmas empresas procurem novas formas de gerir os seus recursos e processos produtivos com uma maior eficiência.

Um sistema de gestão integrado é a solução comprovada para a gestão eficaz dos negócios, pois permite otimizar o desempenho operacional e assegurar o cumprimento da legislação em áreas estratégicas. Um sistema deste tipo tem por base vários referenciais normativos, que auxiliam em várias áreas como, a qualidade, que visa a satisfação, a confiança, a fidelização dos Clientes, o ambiente que procura controlar o impacto ambiental e a segurança e saúde dos trabalhadores que tem como objetivo controlar o risco de acidentes. Contudo, existem muitas outras áreas que estão a ter uma importância crescente por parte das empresas, sendo a manutenção uma das áreas que se destaca, pois acaba por interferir diretamente com as restantes.

Neste sentido, a adoção de um sistema de gestão da manutenção poderá ser uma mais-valia para todas as partes interessadas de uma empresa. A norma NP 4492:2009, especifica os requisitos de um sistema de gestão da manutenção, de modo a que as organizações melhorem o desempenho dos seus processos. Esta norma tem como objetivo certificar prestadores de serviços de manutenção de qualquer dimensão e pressupõe que a atividade da empresa se desenvolve no respeito pela legislação aplicável ao seu sector. A norma NP 4492:2009 impõe uma obrigatoriedade relativamente aos requisitos do Cliente e às exigências legais e regulamentares aplicáveis [32].

As normas portuguesas existentes para a manutenção [25] são as seguintes:

- **NP EN 13306:2007** - Terminologia de manutenção;
- **NP EN 13269:2007** - Manutenção - Instruções para a preparação de contratos de manutenção;
- **NP EN 15341:2009** - Manutenção - Indicadores de desempenho de manutenção;
- **NP EN 13460:2009** - Manutenção – Documentação para manutenção;
- **NP 4483:2009** – Norma guia para a implementação de sistemas de gestão de manutenção;
- **NP 4492:2009** – Requisitos para a prestação de serviços de manutenção.

Com o objetivo de cumprir os requisitos para um sistema de gestão da manutenção, e com todos os avanços da tecnologia ao longo dos anos, cada vez mais os *softwares* de supervisão tem um papel importante nesta área, pois é através destes que toda a gestão de recursos e processos, manutenção de equipamentos, ordens de trabalho e tarefas agendadas, será feita.

Um sistema de supervisão destina-se à capturar e armazenar numa base de dados, informação sobre um processo de produção. A informação é recolhida por meio de variáveis que comunicam com vários equipamentos permitindo a sua leitura e posterior análise do processo. Os sistemas de supervisão são conhecidos como HMI/SCADA (*Interface* Homem Máquina e Controlo de Supervisão e Aquisição de Dados) [26].

Para um processo de produção industrial pode-se citar as seguintes vantagens quando se utiliza um sistema HMI/SCADA:

- **Análise de tendências:** baseado no histórico das informações da base de dados, é possível tomar ações proactivas para maximizar a produção;
- **Alarmes:** sinaliza em tempo real, alguma falha no processo e regista essa falha na base de dados para consultas futuras;
- **Operação Remota no Processo:** permitir intervenção no processo remotamente;
- **Criação de relatórios e gráficos:** É possível criar relatórios e gráficos sobre os alarmes e tendências;
- **Aumentar a eficiência:** A partir das informações geradas em tempo real permite identificar falhas e consequentemente otimizar as tomadas de decisão para manter o processo de operação.

Alguns sistemas de Supervisão em destaque no mercado são:

A. Eclipse SCADA, da *Eclipse Software*;

O Software Eclipse SCADA [27], produto da Eclipse Software, empresa brasileira sediada em Porto Alegre é um *software* de supervisão completo. Permite a visualização de variáveis, utilização de animações, programação de *setpoints*, controlo de acessos e funções especiais para *touchscreen*. O Eclipse SCADA apresenta-se como uma forma muito simples para criar, organizar e monitorizar processos. A criação da *interface* com o usuário é feita de maneira

simples e rápida. Possui vários recursos visuais como animações, displays, botões e gráficos ligados diretamente às variáveis de campo. É um *software* de baixo custo comparativamente a outros concorrentes.

B. IFIX SCADA, da *General Electric*;

O IFIX é um *software* SCADA [29] ideal para aplicações de HMI simples. O *software* suporta uma resolução dinâmica de gráficos, incluindo uma utilização mais eficiente e reduzida da CPU.

C. INDUSOFT WEB STUDIO, da *InduSoft*;

O *InduSoft Web Studio* [30] é um *software* SCADA da *Indusoft* com um vasto conjunto de ferramentas de automação que permitem a construção e desenvolvimento de HMIs e sistemas SCADA. Este *software* permite desenvolver uma aplicação e implantá-la em qualquer plataforma que suporte *Microsoft*, incluindo o *Windows CE* e *Mobile, XP*. Também possui uma alta flexibilidade, permitindo a comunicação a qualquer controlador.

D. SIMATIC Wincc , da *Siemens*;

O *Simatic WinCC* [31] é um sistema SCADA da *Siemens* que se baseia na tecnologia *Windows*, o que proporciona ambientes multitarefas e multiutilizador. Como é um sistema aberto, o *WinCC* permite a expansão do próprio sistema e a interligação com aplicações de terceiros. É um *software* bastante utilizado em vários tipos de indústria e apresenta uma eficiência elevada.

Apenas a título de exemplo serão descritos alguns dos protocolos de comunicação permitidos por cada uma das marcas apresentadas, os protocolos apresentados nas listas seguintes não são os únicos protocolos de comunicação suportados pelas marcas mas são dos mais conhecidos e utilizados.

E. VIJEO CITECT, da *Schneider Electric*;

O *Vijeo Citect* [28] é um *software* de supervisão e controlo de processos e está totalmente integrado em sistemas HMI/SCADA. O *Vijeo Citect* foi concebido por engenheiros de controlo, gerentes de produção e integradores de sistema, que procuravam um SCADA flexível e totalmente integrado nos equipamentos *Schneider*. O *Vijeo Citect* é um SCADA aberto com uma vasta gama de drivers de protocolo para se comunicar com outros dispositivos e controladores.

F. *WONDERWARE INTOUCH*, da *Invensys*;

Os *Human-Machine Interfaces* (HMIs) e soluções SCADA variam muito de simples e diretos, para o complexo e exigente. O *InTouch* [4] [5] é um *software* de visualização, juntamente com o *System Platform*, que se encontram estrategicamente posicionados para serem extremamente fáceis de usar, bem como poderosos e sofisticados para atender a mais exigente e desafiante solução.

Soluções desenvolvidas a partir de *ArchestrA*, ambiente gráfico utilizado pelo *System Platform* [6] em desenvolvimento, beneficiam da tecnologia a partir de uma arquitetura de *software* aberta e escalonável que pode comunicar praticamente com qualquer sistema de automação, unidade terminal remota (RTU), dispositivo eletrónico inteligente (IED), controlador lógico programável (PLC), base de dados ou processador de histórico. Uma mais-valia neste *software* relativamente a outros é o facto da natureza aberta da plataforma, permitir que os usuários possam expandir os seus sistemas existentes sem ter que comprar novos sistemas de hardware ou de controlo.

No Capítulo 3 são mostradas todas as potencialidades dos *softwares Wonderware* e os seus potenciais usos em diversas indústrias.

A solução *InTouch* foi a escolhida para o desenvolvimento desta aplicação, pois é o *software* SCADA utilizado para desenvolvimento de HMIs na SA – Soluções em Automação S. A, sendo a empresa a distribuidora a nível nacional deste *software*.

3. SOLUÇÃO WONDERWARE

A *Wonderware* [11,16] é a líder global em *software* de gestão de operações HMI, SCADA e em tempo real. As soluções *Wonderware* permitem otimizar operações industriais e de produção, de forma a sincronizar com os objetivos de negócio alcançando uma maior velocidade, flexibilidade e rentabilidade sustentada. Os *softwares Wonderware* oferecem benefícios significativos de custos para a concepção, construção, implantação e manutenção de aplicações robustas para operações de produção e infraestrutura [22].

A *Wonderware* cria soluções de *software* para sistemas de gestão de operações, incluindo uma linha de produtos que ajudam a controlar os dispositivos de automação industrial. Em indústrias como a química, gás, energia, mineração, alimentos e bebidas fazem uso dos produtos da *Wonderware*. Entre os vários produtos da empresa, são programas de *software* da *Wonderware InTouch* HMI com interfaces gráficas, que permitem aos operadores comunicar com sistemas automatizados. Nos subcapítulos seguintes serão abordados alguns desses *softwares*.

3.1. *Wonderware InTouch*

Durante 25 anos o *InTouch* da *Wonderware* tem sido um *interface* homem-máquina competente, ou seja, um interface HMI que oferece inovação, excelentes gráficos, facilidade de utilização, conectividade insuperável, um melhor suporte no mercado e dos mais abrangentes ecossistemas da atualidade [4,11].

O *InTouch* fornece uma visualização gráfica que permite a gestão de operações de controlo e otimização para um nível completamente novo. O que a indústria entende hoje como HMI, começou há 25 anos atrás com este *software*, estando em constante inovação e apresentando uma inigualável integração de dispositivos, conectividade e verdadeira facilidade de uso [11].

O *software InTouch* é uma solução de supervisão HMI e SCADA, aberta e extensível, que oferece uma maneira rápida e simples de criar aplicações HMI reutilizáveis e padronizadas, para depois empregá-las em toda empresa sem ter que sair do escritório. Estas aplicações são criadas para sistemas operativos *Microsoft*. Aplicações *InTouch* são usadas em toda a parte do mundo, na fabricação de todo o tipo de produtos, incluindo o processamento de alimentos, semicondutores, petróleo, gás a papel por exemplo.

A monitorização ocorre por meio de símbolos gráficos de alta qualidade, com indicações dinâmicas do estado dos equipamentos e grandezas analisadas. Possui também ferramentas para realizar gráficos de tendência, chamados *Trends*, relatórios, histórico de eventos e de processo, entre outras. Assim como a monitorização, a operação é também realizada com o recurso a símbolos gráficos, com possibilidade de execução de comandos e com o uso de objetos animados extremamente realistas. Estes representam com a mais alta perfeição os equipamentos de acionamento físico, normalmente encontrados em qualquer processo. Entre outras

características, garante a segurança do processo ao permitir atribuir níveis de acesso antes de qualquer comando, assim, exige que os operadores se identifiquem, por meio de palavra-passe para executar qualquer comando. Na Figura 3.1 é mostrado um pequeno exemplo de uma aplicação criada em *InTouch*, com o objetivo de controlar um tanque de água.

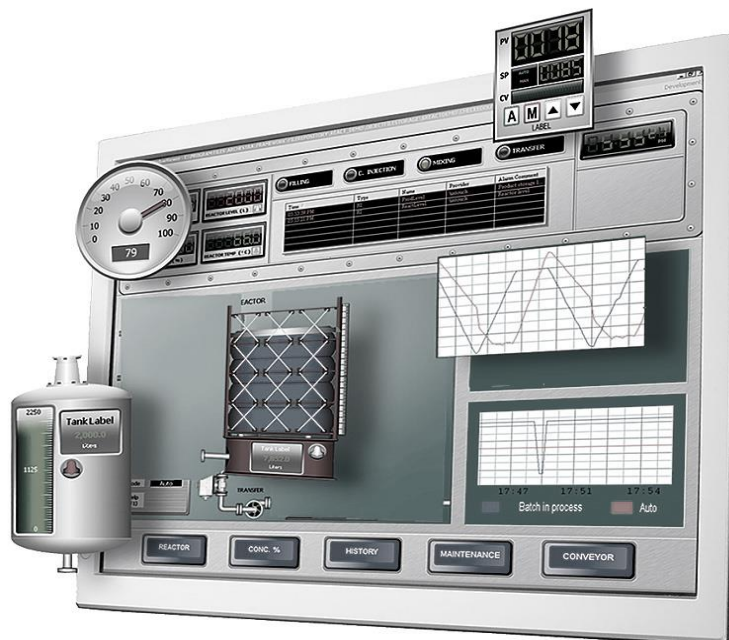


Figura 3.1: Exemplo de uma aplicação realizada em *Wonderware InTouch*.

O *InTouch* pode ser utilizado como solução para, entre outras, suprir as seguintes necessidades das empresas [22]:

- Tornar a operação de um processo ou de uma máquina mais fácil e intuitiva;
- Atribuir maior segurança para os operadores, por meio de operação à distância, sem ter de despender grandes quantidades de tempo e dinheiro em formações;
- Garantir maior controlo e segurança do processo, impedindo que pessoas ou operadores executem operações não autorizadas, inadvertidamente;
- Tornar o seu produto mais atrativo e competitivo no mercado;
- Implementar um sistema completamente expansível, que possibilite acompanhar o crescimento do seu negócio;
- Um sistema com ótima relação custo/benefício;
- Um sistema que permita não depender de um único fornecedor, que tenha uma base instalada significativa e que tenha diversas empresas locais capacitadas a integrá-lo.

Quais são as principais características técnicas [11]?

A. **Archestra Graphics**: Um novo conceito de objetos gráficos é aplicado no *InTouch*. Além da qualidade gráfica dos objetos, possui uma linguagem totalmente orientada ao objeto, com alarmes, variáveis, lógicas, e outros componentes incorporados. Os objetos são

completamente vetoriais, o que permite serem utilizados em qualquer tamanho ou resolução;

- B. **Aquisição de dados:** O *InTouch* HMI incorpora uma grande quantidade de *drivers* de comunicação, o que permite comunicar com a maioria dos autómatos comerciais existentes no mercado. Permite ainda comunicar simultaneamente com mais de um equipamento ao mesmo tempo, mesmo quando forem de fabricantes e protocolos diferentes;
- C. **Histórico:** Permite efetuar o armazenamento de informações do processo ao longo do tempo. Este trabalho pode ser potencializado em plataforma SQL, para sistemas com maiores necessidades, com o uso do *Historian*;
- D. **Gráficos:** Pode-se visualizar gráficos de tendência do processo em tempo real, ou relativo ao histórico. Pode-se também potencializar esta função, com o uso do *Historian*;
- E. **Relatórios:** Está disponível com o *InTouch*, a possibilidade de criar relatórios com as informações monitorizadas pelo sistema;
- F. **Alertas:** O *InTouch* permite a definição de alarmes para condições que representam qualquer risco. Estas condições necessitem de alguma ação do operador ou servem apenas para alertá-los sobre alguma situação específica;
- G. **Scripts:** Permite criar procedimentos baseados em análise condicional, permitindo ao sistema a tomada de decisões;
- H. **Segurança:** Permite definir utilizadores e níveis de acesso ao sistema, tornando-o totalmente seguro e protegido de utilizadores não autorizados;
- I. **Interatividade:** Permite interagir diretamente com outros sistemas ou com outras estações de *InTouch*;

Nas modernas instalações industriais de hoje, existe uma infinidade de fontes de dados, isto é, dispositivos de campo com capacidade de comunicar diretamente com *Programmable Logic Controllers (PLCs)* e com Sistemas de Controlo Distribuído (*DCS*) plenamente desenvolvidos. O *InTouch* é líder em sistemas abertos e conquistou a sua reputação por possibilitar a conexão a mais dispositivos e sistemas do que qualquer outro [11,16].

O *InTouch* pode comunicar com centenas de servidores *I/O* e *OPC* disponíveis, e a ferramenta *ArchestrA DA Server* permite que facilmente se crie servidores de dados especializados, se necessário.

3.2. Wonderware System Platform

O *System Platform* [11] é uma plataforma de aplicação industrial estratégica que é construída com base em tecnologia *ArchestrA*.

O *System Platform* foi criado e pensado com o objetivo de, entre outros, gerir as necessidades de um processo de automação industrial e informação pessoal. Esta solução fornece uma plataforma única e escalonável para toda a supervisão SCADA, geograficamente distribuídas (Geo-SCADA), HMI, de Produção e Gestão de Desempenho e pode ser aplicada

independentemente do tamanho do processo, possuindo características como: plataforma descentralizada, configuração e manutenção simples e fácil adequação às novas necessidades.

Esta plataforma da *Wonderware*, representa uma unidade de negócio da *Invensys*, sendo projetada para tornar mais fácil aos fabricantes e operadores industriais ajustarem as suas necessidades consoante a mudança dos Clientes e do mercado abrangente.

O *System Platform* contém um conjunto de serviços (Figura 3.2) que suportam melhorias sustentáveis nas áreas de produção, operações e desempenho. Na sua essência, o *System Platform* facilita a fabricação consistente e confiável, assim como operações industriais, protegendo sempre a integridade da marca. Este permite que os utilizadores possam estender os seus sistemas em praticamente qualquer direção para atender as necessidades correntes e futuras. Oferece serviços de domínio industrial que não são fornecidos por sistemas de operação comerciais, fornecendo também uma poderosa infraestrutura que possibilita aos Clientes *Wonderware*, mesmo com um *hardware* comercial de baixo custo alcançarem os seus objetivos em termos de aplicações industriais [6].

Aplicado em indústrias de diversos segmentos por todo o mundo, pode ser implementado conforme as necessidades de cada tipo de processo [22]. Associado a poderosas ferramentas de base de dados, analisa e grava grandes quantidades de informações em curtos espaços de tempo, possibilitando assim, obter o máximo desempenho na tomada de decisões.

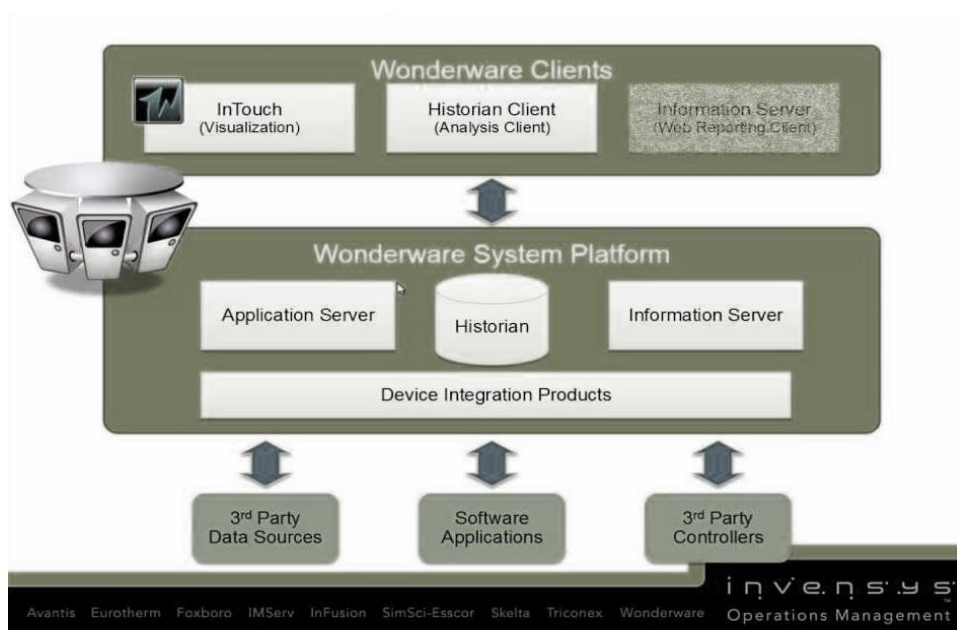


Figura 3.2: Diagrama funcional do *System Platform* relativamente aos softwares da *Wonderware*.

O *System Platform* é uma infraestrutura de aplicação estratégica que capacita empresas para tirar o máximo partido das novas oportunidades de negócios. Fá-lo, permitindo identificar abordagens inovadoras, novas técnicas de gestão de produção, de informação e a possibilidade de criar diferentes tipos de aplicações de *software* para satisfazer as suas necessidades mais específicas [6].

O *System Platform* pode ser utilizado como solução para suprir as seguintes necessidades das empresas [22]:

- Possuir um método de gerir um sistema automatizado, com baixo custo de manutenção e engenharia;
- Implementar ferramentas de aquisição e análise de informação mais detalhada, do seu processo de produção em tempo real, a fim de tomar decisões em menor tempo, tornando-o mais competitivo;
- Implementar um sistema automatizado por etapas, no qual pode agregar novas funcionalidades, aumentando assim, a abrangência do sistema conforme a necessidade e disponibilidade momentânea;
- Uma solução que auxilie a equipa de engenharia a gerir um sistema de automação de grande porte, minimizando os riscos de erros de implementação, falhas e conflitos nos projetos e problemas de compatibilidade;
- Uma solução que permita comunicar com outros sistemas pré-instalados ou sistemas de controlo dedicados de determinadas máquinas;
- Uma solução de alta disponibilidade, com características funcionais que auxiliam a reduzir a vulnerabilidade do processo. Isto, com o objetivo de manter a operação e aquisição de dados 24 horas por dia, 7 dias por semana, mesmo em caso de falhas.

O *System Platform* e a sua ferramenta de *design ArchestrA* fornecem um desenvolvimento fácil e intuitivo de soluções de *software* industrial modular, que podem ser facilmente alteradas para atender necessidades futuras dos Clientes. Como resultado, pode-se criar e gerir padrões de desenvolvimento de aplicações de uma só vez. O resultado é uma diminuição na quantidade de tempo e custos associados com a criação, modificação, implantação, manutenção e padronização de aplicações de *software*.

Quais são as principais características técnicas?

- A. **Aquisição de dados:** Em tempo real, comunica-se com diversos equipamentos de controlo existentes no mercado, disponibilizando a informação para todos os sistemas por ele geridos, sem necessidade de ferramentas complementares;
- B. **Segurança:** Uma extensão do modelo de segurança *Microsoft*, fácil de configurar e que pode ser atribuído a todas as partes do sistema;
- C. **Gestão distribuída de Alarmes:** Avançadas ferramentas de gestão de alarmes, distribuídos por todo o sistema, facilitando a visualização e análise dos mesmos, em diferentes pontos do sistema;
- D. **Ferramenta de desenvolvimento:** Permite desenvolver toda a aplicação orientada ao objeto, possibilitando assim visualizar facilmente todo o processo, efetuar alterações e manutenção em menos tempo e com menores erros. Também possui suporte em linguagem *.NET*, *Visual Studio® development System*, *SQL Server*, *BizTalk® Server*, *SharePoint® Capabilities*, *Microsoft Office* e *Internet Explorer® Internet browser*;

- E. **Conectividade:** Ferramentas de conexão à base de dados SQL, *Oracle*, entre outros. Comunicação em *OPC* e ferramentas de desenvolvimento de *drivers* de comunicação dedicados;
- F. **Ambiente Intuitivo:** *Interface* de fácil compreensão e visualização, com diversas ferramentas de apoio ao desenvolvimento;
- G. **Desenvolvimento remoto:** Distribuição remota de modificações, manutenção e distribuição remota por estação. Fácil redistribuição por mais que um servidor permitindo alterações *online*.

3.3. *Wonderware Historian*

O *Historian* [11] é um histórico de processos de alto desempenho capaz de armazenar fluxos enormes de dados provenientes das mais diversas instalações industriais automatizadas. O *Historian* é um dos mais populares processadores de histórico do mercado, sendo usado em mais de 25.000 instalações industriais. Este *software* da *Wonderware* foi o primeiro grande processador de histórico a combinar um sistema de aquisição de dados de alta velocidade e armazenamento, com o *Microsoft SQL Server*, permitindo assim a existência de maior capacidade de recuperação de dados [8]

Altamente escalonável e flexível, o *Historian* pode ser aplicado a um único processo, ao longo do tempo, ou seja, a uma solução de gestão de informação de múltiplas camadas, capaz de servir dezenas de *sites* e centenas de utilizadores. A sua arquitetura de várias camadas permite um custo-benefício económico mais eficiente, assim como um acesso mais amplo e seguro a dados pelo utilizador, aumentando assim o seu *Return on Investment* (ROI) do processo.

Como um componente central do *System Platform*, o *Historian* possui conectividade com a mais ampla variedade de *data sources* e sistemas industriais como mostrado na Figura 3.3. As suas capacidades avançadas de recuperação de dados, significam que as empresas não irão necessitar de um profissional a tempo integral para gerar relatórios. O *Historian* integra também uma linha completa de *desktop*, *web*, análise de dados móveis e *reporting clients*, possibilitando a visualização e análise de processos a partir de qualquer local.

Grandes vantagens apresentadas pelo *Historian*.

- Histórico de aplicações até 2.000.000 *tags* de dados;
- Alta disponibilidade ou requisitos de recuperação elevados;
- Aplicações que geram dados analógicos, discretos, bem como alarmes e eventos;
- Aplicações que requerem acesso de dados móveis;
- Aplicações com despesas de capital graves (*CAPEX*) ou restrições a nível de Tecnologia da Informação (*TI*).

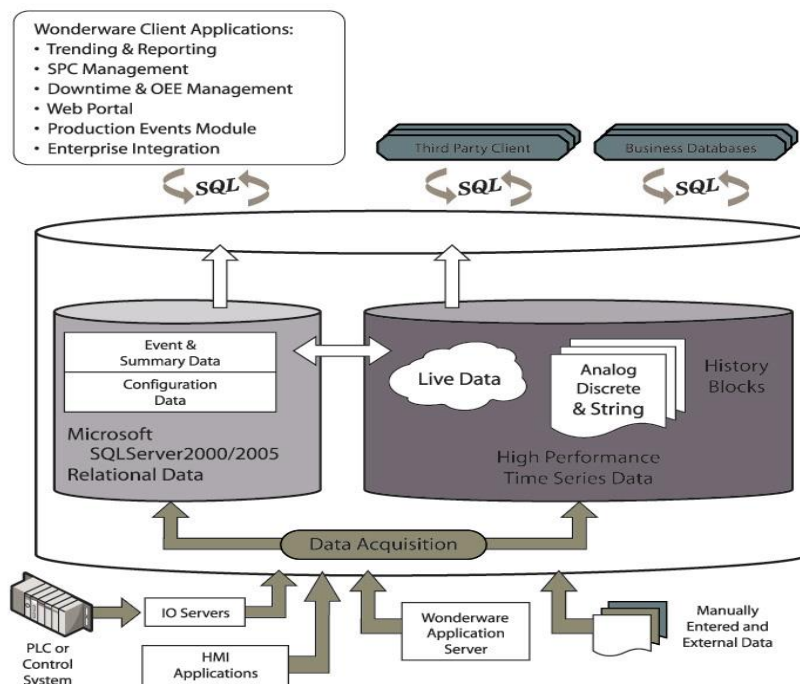


Figura 3.3: Diagrama representando a interatividade do *Historian* [8].

O *Historian* tem desfrutado de uma evolução contínua para melhor atender as necessidades dos seus utilizadores. Nos dias que correm o *Historian* é capaz de armazenar dados de 2 milhões de *tags*, com resolução de milissegundos. Este processador de histórico pode ser configurado para atender às necessidades específicas de indústrias, tais como petróleo e gás, energia, minerais, metais, alimentos, papel e celulose, produtos farmacêuticos e muitos mais [11].

A configuração é simples quando usado com o *System Platform* e o valor do *InTouch* HMI é muito prolongado devido à captura de dados em tempo real, podendo solucionar problemas futuramente, ou otimizar processos. Os dados de histórico vão-se tornar essenciais para as operações de gestão, organização e manutenção. Os trabalhadores da indústria de hoje precisam de informações corretas e objetivas para executarem o seu papel ao melhor nível sendo precisamente isso que o *Historian* oferece.

Como grande ferramenta do *Historian*, tem-se o *Historian Client*, que pode ser incorporado em diversas aplicações, como *Word*, *Excel*, *InTouch*, *Internet Explorer* ou qualquer outro sistema que tenha suporte a *APIs*. Assim, permite tirar total proveito de toda a sua potencialidade em qualquer ambiente, sem a necessidade de projetos complexos ou instalações avançadas. Toda a informação fica disponível em qualquer lugar, seja no portátil do responsável pela produção, na estação de operação instalada na linha de produção da fábrica ou até mesmo no computador de uso diário no gabinete do administrador.

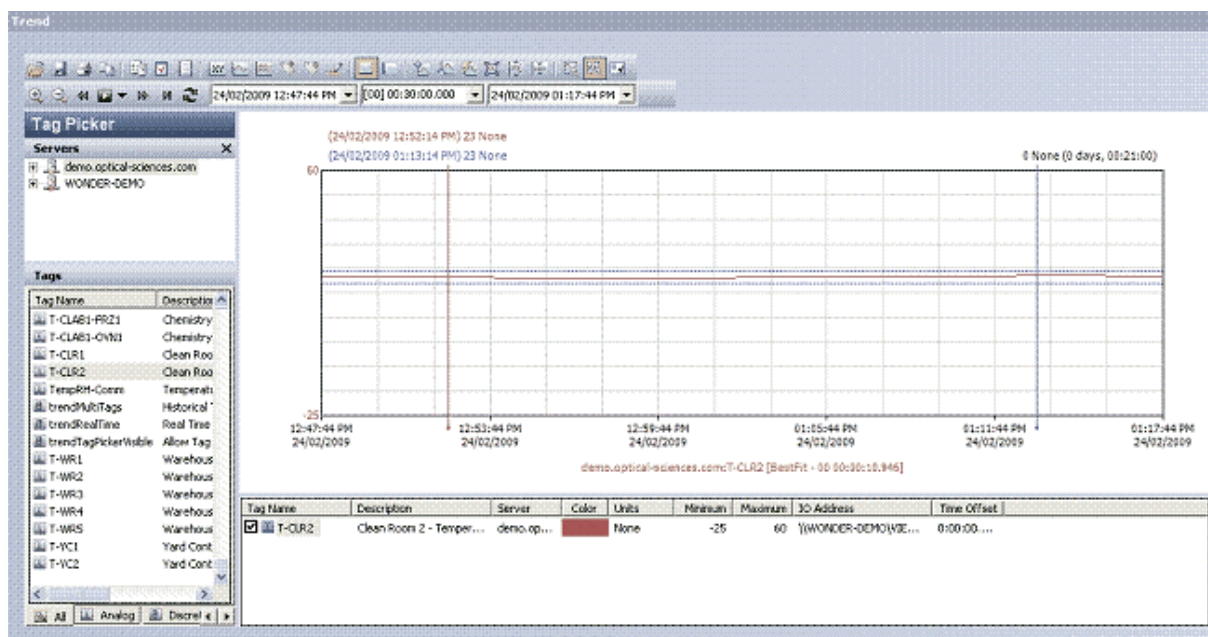


Figura 3.4: Ferramenta *Trend* do *Historian Client* [8].

O *Historian Client* pode ser utilizado como para atender às seguintes necessidades de uma empresa [22]:

- Consultar informações do processo de forma fácil e rápida;
- Visualizar e consultar informações do processo por meio de aplicativos *Word* ou *Excel*, sem precisar de criar macros ou conhecer qualquer tipo de linguagem de programação;
- Visualizar e consultar informações do processo, em qualquer computador ligado à rede, com o uso do *internet explorer*;
- Realizar diversos tipos de consulta, como valores máximos, mínimos, médias, entre outros;
- Visualizar gráficos de tendência e comparar o comportamento de determinados processos com outros períodos;
- Visualizar gráficos de tendência relacionados a dados armazenados em histórico ou em tempo real.

Quais são as principais características técnicas?

A. **Multifunções:** O *Historian Client* incorpora as seguintes funções:

- Uma ferramenta de consultas de dados denominada "*Query*" (Figura 3.4);
- Uma ferramenta de análise de gráficos de tendência denominada "*Trend*";
- Um conjunto de ferramentas de análise de dados para ser incorporado ao *Microsoft Excel* ou *Microsoft Word*;
- Uma ferramenta que disponibiliza toda a informação em ambiente *WEB*;
- *APIs* que permitem utilizar estas ferramentas incorporadas em qualquer sistema que suporte este tipo de funcionalidade.

B. **Gráficos de Tendência:** A ferramenta de análise de gráficos de tendência incorpora as seguintes características:

- Gráfico em tempo real: Pode-se visualizar a informação do processo em tempo real, ou seja, visualizar no gráfico as mudanças que ocorrem no processo, no momento em que ocorrem;
- Valores precisos e análise de período: Pode-se visualizar com total precisão os valores armazenados em função do tempo, e verificar qual a variação de uma medição, num determinado período de tempo;
- Análise de estatísticas: Pode-se visualizar informações estatísticas em relação ao gráfico, como um todo ou num determinado período (por exemplo: valor máximo, mínimo, média, entre outros).

C. **Microsoft Excel:** Ao instalar o *Historian Client*, é instalado automaticamente um suplemento ao *Microsoft Excel*. Este suplemento habilita uma série de ferramentas, disponíveis na barra de ferramentas do *Excel*, o que permite um acesso rápido a consultas. Assim permite criar relatórios instantâneos sem o auxílio de especialistas ou programadores. O mesmo acontece para o *Microsoft Word*.

D. **Query:** A ferramenta *Query*, permite a consulta de forma rápida e direta às informações armazenadas com o *Historian*. Esta ferramenta disponibiliza as informações em formato de tabela, permitindo assim, utilizar os dados apresentados em qualquer programa.

4. SISTEMA DE GESTÃO E MANUTENÇÃO DESENVOLVIDO

Após a análise de necessidades do Sistema de Gestão de Manutenção para uma empresa na indústria cerâmica, facilmente se chegou à conclusão de que existe uma necessidade de criar um sistema de gestão que auxilie o gestor da manutenção no que refere as máquinas utilizadas nos vários processos, equipamentos e material de manutenção da empresa e na gestão dos diversos tipos de ordens de trabalho [3].

Em termos de conceção procurou-se elaborar um modelo de dados que refletisse a realidade da indústria e fosse ao encontro das necessidades a ela inerentes. Numa primeira fase, tentou encontrar-se um modelo em *Microsoft SQL Server*, visto que é o *software* utilizado pela *Wonderware*, capaz de dinamizar todo o processo e o seu respetivo fluxo de atividades, pois só assim seria possível retirar uma ideia daquilo que o sistema iria precisar em termos de *interfaces* [15].

O modelo de base de dados assenta em filosofia SQL, sendo que é aqui que são construídos todos os mecanismos e estruturas necessárias para mecanizar o funcionamento do sistema de gestão e manutenção, utilizando posteriormente o SCADA *Wonderware* para construir todo o *interface* com o utilizador. Na Figura 4.1 encontra-se um diagrama que demonstra as dependências dos *softwares* utilizados na aplicação, bem com as suas interligações em termos de dados a movimentar.

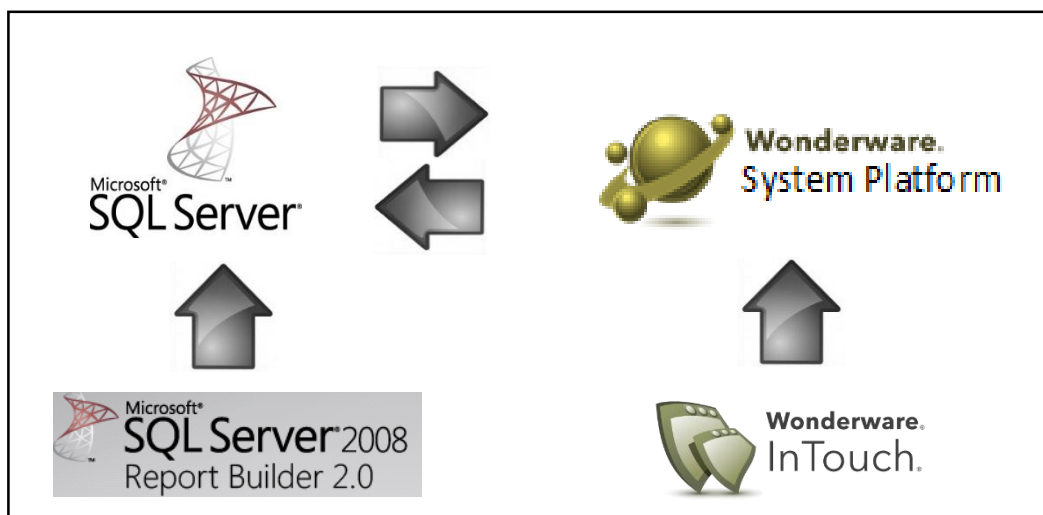


Figura 4.1: Diagrama funcional com as interligações entre os *softwares* utilizados.

Pode-se observar pela Figura 4.1 que tanto o *SQL Server* com o *System Platform*, trocam dados entre si, isto é, utilizam o mecanismo de escrita e leitura em diversos processos. Os relatórios contruídos em *Report Builder* usam conexões ao *SQL* para devolver os dados pretendidos de forma a criar o relatório desejado. O *InTouch* é utilizado para permitir que todos os processos

criados em *System Platform* fiquem funcionais para o operador. O *InTouch* reúne toda informação trabalhada e mostra o resultado final da aplicação.

Nos próximos subcapítulos serão explicados os vários processos de construção e conceção do sistema de gestão e manutenção e as diversas fases do mesmos, isto é, todos os processos desenvolvidos em *SQL Server*, o *interface* gráfico criado em *System Platform*, o sistema de relatórios desenvolvidos em *Report Builder 2.0* e por fim todas as funcionalidades integradas e funcionais no *InTouch HMI*.

4.1. Microsoft SQL Server

A base de dados é composta por uma série de tabelas utilizadas para gravar todo o tipo de dados relativos ao sistema de gestão e manutenção. Com base na experiência no sector por parte dos colaboradores da empresa SA – Soluções em Automação S.A tentou criar-se numa fase inicial uma estrutura capaz de automatizar todo o sistema de gestão, abrangendo equipamentos, equipas, material, agendamentos, ordens de manutenção, de modo a tornar a manutenção o mais eficaz possível. Todo este modelo, que se pode pensar genérico, foi adaptado à realidade da indústria cerâmica atual, com todas as envolventes associadas à mesma.

As várias tabelas construídas, vão ao encontro das necessidades analisadas em estudo na indústria cerâmica, de forma a garantir a melhor solução a implementar, procurando-se incorporar conceitos que possibilitam organizar, estruturar e dinamizar o sistema. Existe interligação entre todas as tabelas, usando chaves estrangeiras. Estas chaves possibilitam a interatividade do sistema e permitem organizar melhor a base de dados, de forma lógica, coerente e intuitiva.

Na Figura 4.2 apresenta-se um diagrama completo da base de dados criada em SQL.

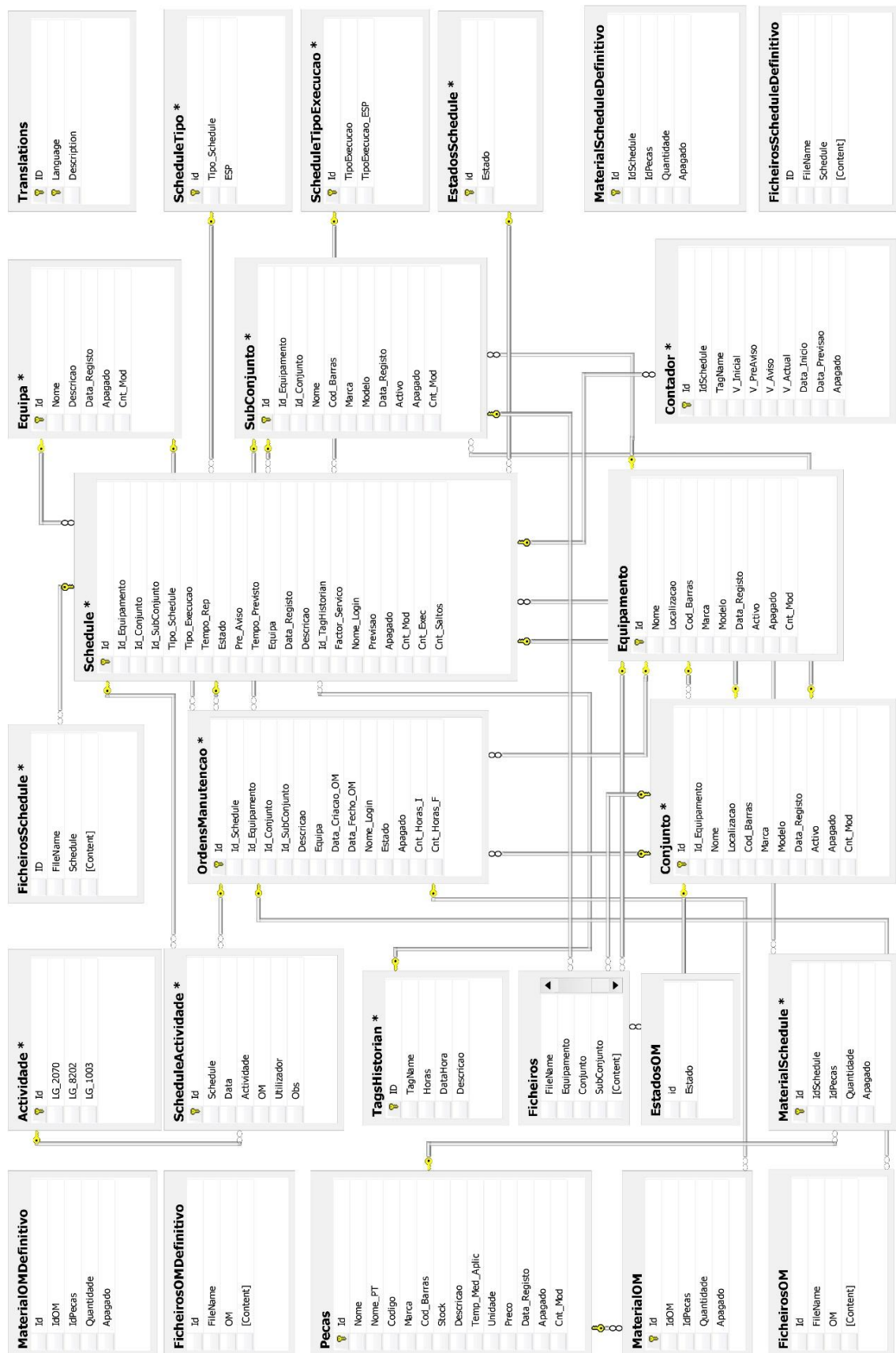


Figura 4.2: Diagrama da base de dados gestão de manutenção desenvolvida.

4.1.1 Tabelas SQL

A. Equipamentos - Conjuntos - Subconjuntos

As tabelas Equipamento, Conjunto e Subconjunto (Figura 4.3) referem-se a todos os equipamentos existente no sistema de gestão e manutenção de uma cerâmica. Neste caso todos os conjuntos encontram-se dentro do equipamento principal e os subconjuntos dentro do respetivo conjunto. Como exemplo, um caso específico de uma cerâmica de tijolo, existe um queimador vertical, onde podem existir vários queimadores que consequentemente têm respetivos bicos associados. Tendo em conta esta filosofia de equipamentos existente nesta indústria optou-se por um modelo mais abrangente em termos de tipo de equipamento utilizado.

Equipamento *	Conjunto *	SubConjunto *
<div> <div>Id</div> <div>Nome</div> <div>Localizacao</div> <div>Cod_Barras</div> <div>Marca</div> <div>Modelo</div> <div>Data_Registo</div> <div>Activo</div> <div>Apagado</div> <div>Cnt_Mod</div> </div>	<div> <div>Id</div> <div>Id_Equipamento</div> <div>Nome</div> <div>Localizacao</div> <div>Cod_Barras</div> <div>Marca</div> <div>Modelo</div> <div>Data_Registo</div> <div>Activo</div> <div>Apagado</div> <div>Cnt_Mod</div> </div>	<div> <div>Id</div> <div>Id_Equipamento</div> <div>Id_Conjunto</div> <div>Nome</div> <div>Cod_Barras</div> <div>Marca</div> <div>Modelo</div> <div>Data_Registo</div> <div>Activo</div> <div>Apagado</div> <div>Cnt_Mod</div> </div>

Figura 4.3: Tabelas SQL Server de Equipamento, Conjunto e SubConjunto.

As variáveis nas colunas das tabelas são exatamente iguais entres todas, destacando-se as respetivas chaves estrangeiras que servem de interligação de equipamentos.

Dos campos utilizados destacam-se:

- **Activo** - indica se o equipamento se encontra em funcionamento na fábrica ou não;
- **Apagado** - permite saber se o equipamento foi removido, caso esteja verdadeiro, o equipamento fica invisível na supervisão mas nunca apagado da base de dados;
- **Cnt_Mod** - Serve de contador permitindo saber quantas vezes os parâmetros do equipamento foram modificados.

B. Equipas

A tabela equipas é importante, pois são as equipas que vão estar associadas a diversos tipos de gestão de processos ou de manutenção.

As variáveis utilizadas são idênticas relativamente aos equipamentos, tendo as características principais das equipas, como a descrição. Os campos destacados anteriormente estão também presentes nesta tabela.

C. Peças

Na tabela Peças, mostrada na Figura 4.4, encontra-se todo o material disponível na base de dados gestão de manutenção. Como o material é um aspeto importante num sistema de manutenção de equipamentos, tomou-se isso em conta ao contruir os campos desta tabela. Não é permitido material com o mesmo nome, mesmo código de peça ou mesmo código de barras.

Dos campos utilizados destacam-se:

- **Código** - indica o código de peça;
- **Tempo Médio de Aplicação** – Representa o tempo médio de instalação da respetiva peça;
- **Data de Registo** – Data de inserção da peça na base de dados.

D. Schedules

Esta tabela de *Schedule* (Figura 4.5) é provavelmente a mais importante e por onde passa grande parte da dinamização da manutenção, sendo de extrema importância para o bom funcionamento de um sistema de gestão, pois é basicamente ela que gere e controla todas as ações a efetuar. É nesta tabela que fica gravada a tarefa agendada de determinada ordem de manutenção, e para essa ordem ser gravada é necessário que todos os parâmetros importantes e inerentes ao agendamento sejam associados a ela. A tabela *Schedule* ou agenda, deve conter os equipamentos a que se refere, de modo a saber-se onde será a intervenção a efetuar.

Como variáveis de colunas importantes nesta tabela podem-se destacar:

- **Tipo de Schedule** - Indica qual o tipo de *schedule* em agenda;
- **Tipo de Execução** - Permite indicar qual o tipo de execução a efetuar, execução única periódica ou infinita;
- **Tempo de Repetição** - Tempo que a *schedule* demora até se repetir novamente;
- **Estado** - Indica qual o estado em que se encontra a *schedule*;

Id
Nome
Nome_PT
Codigo
Marca
Cod_Barras
Stock
Descricao
Temp_Med_Aplic
Unidade
Preco
Data_Registo
Apagado
Cnt_Mod

Figura 4.4: Tabela de Peças em SQL Server.

Id
Id_Equipamento
Id_Conjunto
Id_SubConjunto
Tipo_Schedule
Tipo_Execucao
Tempo_Rep
Estado
Pre_Aviso
Tempo_Previsto
Equipa
Data_Registo
Descricao
Id_TagHistorian
Factor_Servico
Nome_Login
Previsao
Apagado
Cnt_Mod
Cnt_Exec
Cnt_Saltos

Figura 4.5: Tabela *Schedule* da base de dados de gestão e manutenção.

- **Pré-Aviso** -Tempo que a *schedule* demora até atingir o pré-aviso;
- **Tempo Previsto** -Tempo previsto para a execução da *schedule*;
- **TagHistorian** - Contador a que se refere a respetiva *schedule*, podendo ser um contador de funcionamento de máquina ou contador fixo, ou seja contador de relógio normal;
- **Factor de Serviço** - Fator de serviço a que está a funcionar a máquina;
- **Contador de Execuções** - Indica o número de execuções que se pretenda para a respetiva tarefa;
- **Contador de Saltos** - Indica o número de vezes que a *schedule* não foi efetuada e reagendada para a data seguinte.

E. Estado da *Schedule* -Tipo de *Schedule* - Tipo de Execução

As três tabelas da Figura 4.6 estão diretamente relacionadas com a tabela *Schedule*, pois servem de configuração para os respetivos parâmetros da mesma. Assim pode verificar-se quais os estados, tipo de *schedule* e tipo de execução que existem, podendo sempre futuramente modificar processos com o objetivo de melhorar e dinamizar melhor o sistema. A tabela *Schedule* possui respetivamente chaves estrangeiras para cada uma destas tabelas, possibilitando assim uma melhor configuração da *schedule*.

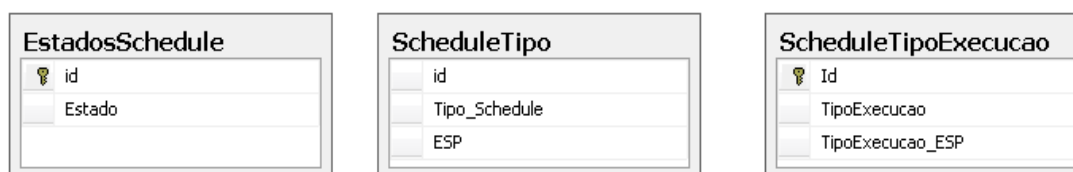


Figura 4.6: Tabelas referentes aos Estados da *Schedule*, Tipo de *Schedule* e Tipo de Execução em SQL Server.

F. Ordens de Manutenção

A tabela Ordens de Manutenção representa todas as intervenções de manutenção efetuadas na gestão do sistema. Estas ordens podem ser originárias diretamente da tabela *Schedule*, sendo configuradas automaticamente com parâmetros da respetiva *Schedule*, ou podem ser ordens independentes, por necessidade momentânea. Nestas últimas, deve indicar-se qual o equipamento a realizar a intervenção, a equipa e o utilizador que registou o evento. Exatamente como na tabela *Schedule* existem campos semelhantes que permitem o correto funcionamento do sistema.

Entre vários campos existentes nesta tabela e como podem existir diferentes estados para uma ordem de manutenção, esta é configurada com a data de registo e data de fecho, aquando da conclusão. As ordens de manutenção podem ser canceladas ou concluídas para ambos os casos, existindo variáveis na tabela que permitem saber quanto tempo esteve a ordem em execução. Tal como acontece com as *schedules*, uma ordem de manutenção nunca é apagada do sistema,

ela é simplesmente removida, ficando invisível na supervisão, mas nunca apagada da base de dados.

A Figura 4.7 apresenta a tabela construída em SQL Server para as Ordens de Manutenção.

OrdensManutencao	
	Id
	Id_Schedule
	Id_Equipamento
	Id_Conjunto
	Id_SubConjunto
	Descricao
	Equipa
	Data_Criacao_OM
	Data_Fecho_OM
	Nome_Login
	Estado
	Apagado
	Cnt_Horas_I
	Cnt_Horas_F

Figura 4.7: Tabela de Ordens de Manutenção do sistema de gestão e manutenção.

G. Estados das Ordens de Manutenção

A tabela de estados serve para descrever os diferentes estados de uma ordem de manutenção. No momento de criação de uma ordem ela é colocada em curso, sendo posteriormente possível mudar o seu estado para *standby*, esperando assim que seja retomada a sua execução. É possível também cancelar ou fechar uma ordem de manutenção, ficando estas inativas mas sempre gravadas na base de dados para estatísticas, caso seja necessário.

H. Material da *Schedule* - Material das Ordens de Manutenção

Estas tabelas (Figura 4.8) permitem que seja associado material à respetiva *schedule* ou ordem de manutenção. Com elas o operador terá a possibilidade de ao criar uma *schedule* ou ordem, pode imediatamente associar material, confirmando depois o processo. Ao modificar, quer uma *schedule*, quer uma ordem de manutenção, também será possível acrescentar ou remover material, caso o operador assim o entenda. É importante referir que ao transformar uma *schedule* em ordem de manutenção, o material que lhe estiver associado, fica automaticamente disponível na tabela respetiva. Estas tabelas funcionam como tabelas temporárias na base de dados, isto é, o operador ao criar ou modificar um agendamento ou ordem, o material é inserido no caso da primeira opção, ou lido no caso de uma modificação, sendo nestas tabelas que fica armazenado esse material. O operador tem então a possibilidade de confirmar toda a operação ou cancelar. Caso confirme todo o material é gravado para uma tabela definitiva de material, quer seja para uma *schedule* ou ordem de manutenção, no caso de cancelamento, nada é gravado

para a tabela definitiva. O conteúdo destas tabelas é sempre apagado depois de qualquer operação, garantindo sempre a correta movimentação do material.

MaterialSchedule	
Id	
IdSchedule	
IdPecas	
Quantidade	
Apagado	

MaterialOM	
Id	
IdOM	
IdPecas	
Quantidade	
Apagado	

Figura 4.8: Tabelas de material para as *schedules* e ordens de manutenção.

I. *Schedule*Atividade – Atividade

A tabela de *Schedule*Atividade, permite saber toda a atividade de uma *schedule* desde a sua criação. A *schedule* pode passar por diversos processos e todos eles podem ser visualizados e analisados nesta tabela, funcionando como um *logger* e permitindo ao operador analisar o progresso das diversas tarefas em agenda. A tabela Atividade possui as várias atividades existentes para um agendamento e interliga-se diretamente com a primeira através de uma chave estrangeira. Estas duas tabelas encontram-se na Figura 4.9.

As atividades consideradas foram:

- Criação da *Schedule*;
- Modificação de Parâmetros na *Schedule*;
- *Schedule* Apagada;
- *Schedule* Entrou em Pré-Aviso;
- *Schedule* Entrou em Aviso;
- Ordem de Manutenção Criada;
- Ordem de Manutenção Modificada;
- Ordem de Manutenção Fechada;
- Ordem de Manutenção Cancelada;
- Ordem de Manutenção Registada para Data Seguinte;
- Ordem de Manutenção Saltada. Fim de Programação de *Schedule*.

Existe uma variável de extrema importância nesta tabela, sendo ela a variável de observações. Sempre que exista criação de uma *schedule*, neste campo são colocados todos os parâmetros importantes configurados na mesma, sendo que se posteriormente houver alguma alteração, pode-se verificar em que campo ocorreu essa mesma alteração. Esta variável serve de auxílio para o operador saber quais os parâmetros que a *schedule* possui e se estes parâmetros foram modificados ou não.

ScheduleActividade *	Actividade *
<div> <div>Id</div> <div>Schedule</div> <div>Data</div> <div>Actividade</div> <div>OM</div> <div>Utilizador</div> <div>Obs</div> </div>	<div> <div>Id</div> <div>LG_2070</div> <div>Actividade</div> <div>LG_8202</div> <div>LG_1003</div> </div>

Figura 4.9: Tabelas *ScheduleActividade* e *Actividade* da base de dados.

J. Contadores

Nesta tabela de Contadores, todo o preenchimento é automático, ou seja, nela são guardados os dados do contador associado à *schedule*, podendo ser um contador de funcionamento de máquina ou contador fixo, que como dito anteriormente funciona como um contador de relógio normal. Um contador é inserido na tabela imediatamente após a criação de uma *schedule*, sendo que os parâmetros de tempo de repetição, pré aviso e fator de serviço da mesma, influenciam diretamente os cálculos dos campos desta tabela. Quando acontece uma modificação de uma tarefa em agenda, o contador pode também sofrer alterações caso os parâmetros referidos anteriormente forem alterados. Esta tabela é bastante importante, porque é ela que permite saber quando uma determinada *schedule* entrou ou ultrapassou o aviso, sendo urgente prosseguir com a sua ordem de manutenção.

K. TagsHistorian

Os contadores existentes no sistema estão guardados nesta tabela. Ela está diretamente ligada ao *Historian* e está sempre em contagem, logo os contadores não param. Aqui pode consultar-se o número total de horas do contador, por exemplo. A tabela de Contadores, relaciona-se com a *TagsHistorian*, de modo a saber qual o contador a associar à tarefa em agenda e qual o seu valor na altura da inserção, contabilizando o funcionamento da máquina a partir desse instante.

L. Ficheiros Equipamento - Ficheiros *Schedule* - Ficheiros OM

Nas tabelas de Ficheiros encontra-se toda a documentação inerente ao sistema de gestão e manutenção. Nestas tabelas (Figura 4.10) são guardados todos os ficheiros, sejam eles word, pdf, jpeg, wordpad, txt, relativamente ao equipamento pretendido, *schedule* ou ordem de manutenção respetivamente. O ficheiro em si, é guardado em formato binário contendo o nome, caminho e extensão do mesmo, podendo mais tarde abrir-se qualquer ficheiro a partir da base de dados SQL. Como vantagem, os ficheiros deixam de estar guardados em pastas ou diretorias próprias, passando a estar na respetiva tabela com informação inerente aos respetivos equipamentos, *schedules* ou ordens.

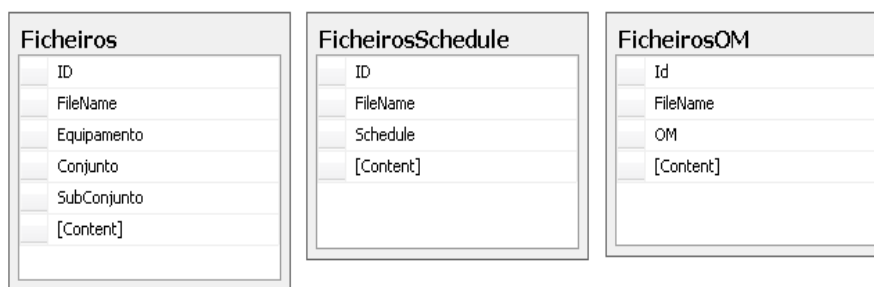


Figura 4.10: Tabelas onde é guardada toda a documentação inerente ao sistema.

4.1.2 Views SQL

Um *View* [1] em SQL pode ser pensado como uma tabela virtual ou uma consulta armazenada. Os dados acessíveis através de um *View* não se encontram armazenados na base de dados como um objeto distinto, o que é armazenado, é uma instrução *SELECT*. O conjunto de resultados da instrução *SELECT*, forma a tabela virtual retornando o resultado pretendido. Um utilizador pode usar essa tabela virtual referenciando o nome do *View* em instruções *Transact-SQL*, da mesma forma que uma tabela é referenciada.

Utilizando o *View* como uma ferramenta essencial na visualização de dados, a informação presente em todas as tabelas da base de dados foi analisada e filtrada, com o objetivo de a tornar mais acessível e direta ao utilizador. Foram construídos variados *Views*, permitindo todo o tipo de cruzamento de dados de quaisquer tabelas, possibilitando assim a visualização da informação requerida, apresentando uma enorme vantagem para uma gestão e manutenção cuidada e objetiva do sistema.

Relativamente aos equipamentos, foram criados diversos *Views* que permitem ao utilizador apenas visualizar os equipamentos que não estejam apagados, ou seja, em que a variável Apagado se encontre a falso. De forma análoga existem *Views* que repetem a mesma lógica nas tabelas de Peças, Equipas, Material, *Schedules* e Ordens de Manutenção.

Para as *Schedules*, é possível visualizar todas as existentes com todos os campos associados e informação mais clarificada. Existem *Views* que permitem ver quais as *schedules* que estão somente em agenda, em pré-aviso, aviso, ou que estão com ordem de manutenção. Uma mais-valia é o facto de estes procedimentos apresentarem campos calculados, e podendo verificar-se quantas horas faltam para atingir um pré-aviso ou aviso, ou mesmo quantas horas passaram desde que a *schedule* entrou em aviso. Estes dados são bastante importantes para a gestão de processos porque permitem informar com objetividade o estado em que se encontram os agendamentos.

Consultando estes procedimentos, pode-se tomar decisões mais concisas quanto ao caminho a seguir e as intervenções a tomar para uma manutenção melhorada.

Na Figura 4.11 mostra-se como é feita a construção de um *View*, onde é mostrada a interligação entre várias tabelas do sistema, a informação pretendida e a instrução *SELECT*, onde é filtrada

a informação a visualizar. O resultado da instrução é dado em formato de colunas com os respectivos dados requeridos. O *View* apresentado refere-se a todas as *schedules* que estão agendadas, apresentando as características das mesmas.

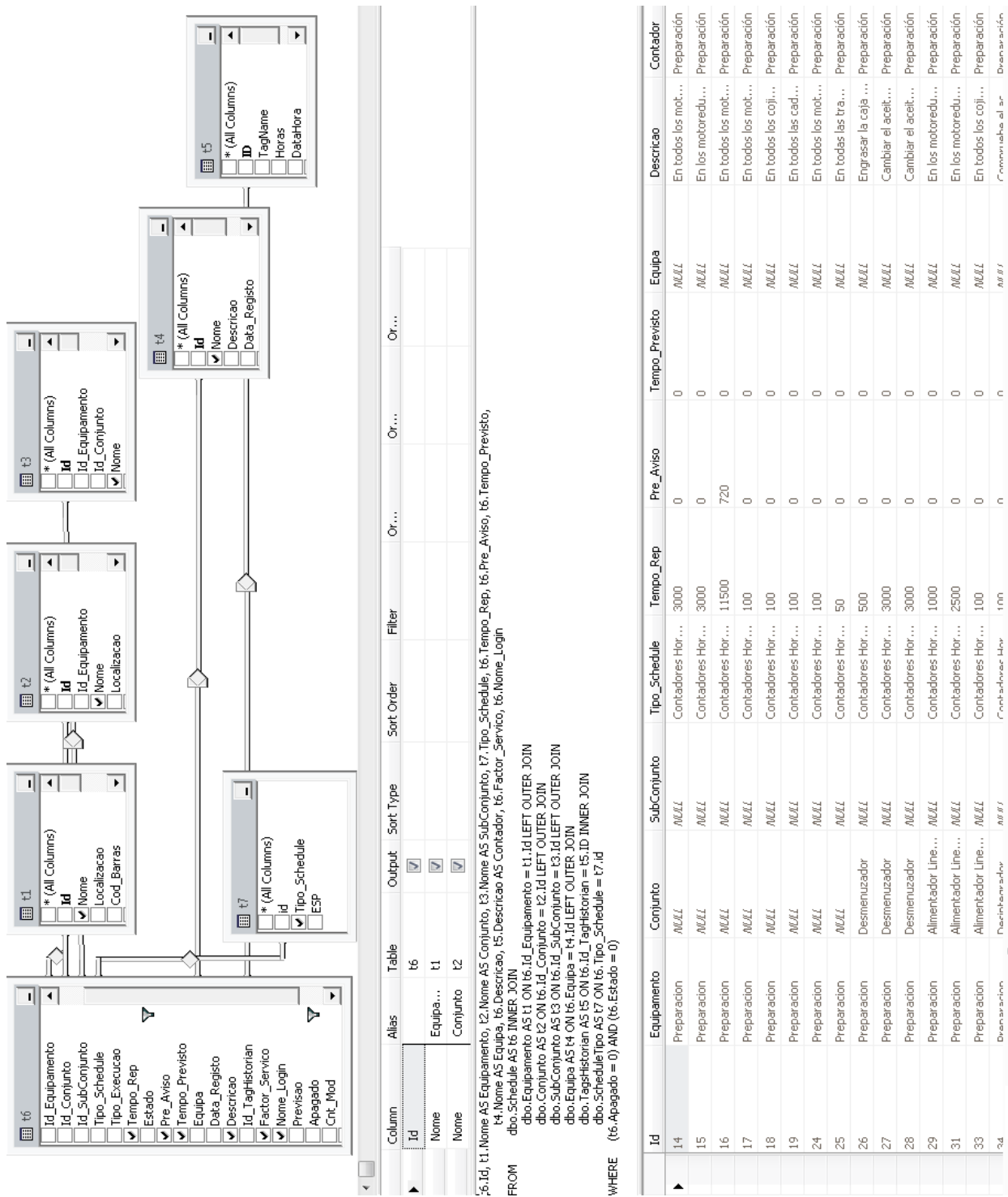


Figura 4.11: *View* SQL apresentando todas as *schedules* em agenda.

4.1.3 *Stored Procedures* SQL

Um *stored procedure* [1] é um procedimento armazenado que tem como objetivo, ajudar a alcançar uma implementação coerente e lógica entre aplicações. É uma coleção de comandos em SQL que engloba tarefas repetitivas, aceita parâmetros de entrada e retorna um valor, indicando o sucesso na execução, ou falha.

As instruções e a lógica SQL necessária para realizar uma tarefa comumente executada, pode ser projetada, codificada e testada uma só vez, utilizando um *stored procedure*. Cada aplicação que tenha a necessidade de efetuar essa tarefa pode simplesmente executar o procedimento, garantindo o correto funcionamento de todo o processo a ele associado.

Os *stored procedures* têm a grande vantagem de poderem melhorar o desempenho do sistema, pois a lógica condicional aplicada aos resultados das primeiras instruções SQL, determina que instruções subsequentes sejam executadas. Sendo estas instruções e lógica condicional escrita num procedimento armazenado, tornam-se parte de uma execução única do servidor, acelerando todo o processo, para além, de melhorar consideravelmente a performance, criando mecanismos de segurança entre a manipulação dos dados no sistema.

Estes procedimentos foram usados para o preenchimento de todas as tabelas da base de dados e não só. A interação com o operador é bastante importante neste tipo de soluções, e tendo isso em mente, os *stored procedures* são bastante importantes para permitirem a interligação com o *System Platform*, possibilitando uma solução SCADA dinâmica e objetiva.

Para cada tabela existente no sistema existem procedimentos que permitem o seu preenchimento. Procedimentos de Adicionar, Modificar Apagar, Remover e Ler, servem todas as tabelas, ou seja, em todas elas existem *stored procedures* deste tipo, adaptando-se como é lógico aos campos apresentados por cada uma. Passa-se a descrever cada um destes procedimentos em mais pormenor:

- A. **Procedimento Adicionar:** Permite adicionar elementos à tabela com todos os parâmetros associados à mesma, devolvendo sempre o número ou *ID* do elemento inserido:
- B. **Procedimento Apagar:** Permite apagar da base de dados um elemento inserido incorretamente ou por engano, estando sempre associado ao procedimento de Adicionar. Por exemplo, pretende-se adicionar um certo equipamento na base de dados, e no momento da sua inserção e preenchimento de parâmetros resolve-se cancelar a operação, nesse momento o procedimento apagar é chamado, apagando a linha que se estava a inserir;
- C. **Procedimento Remover.** Este procedimento é feito exclusivamente a pensar na supervisão, pois quaisquer elementos que se pretendem apagar, é este procedimento que o faz, mas na verdade não o apaga da base de dados. As instruções SQL permitem escolher um elemento de qualquer tabela e ao remover, está-se a mudar o estado de um bit que faz com que na supervisão o elemento não esteja visível, embora os dados estejam sempre disponíveis na base de dados:

- D. **Procedimento Modificar:** Permite escolher qualquer elemento de uma tabela e modificar os parâmetros permitidos. Este procedimento é utilizado quando se pretende modificar parâmetros específicos em algum determinado tipo de dados, por exemplo modificar parâmetros de uma *schedule*. Na Figura 4.12 pode-se ver o programa utilizado para realizar este processo de modificar campos;

```
ALTER PROCEDURE [dbo].[EquipamentoUpdate]
(
    @Id          INT,
    @Nome        VARCHAR(50),
    @Localizacao VARCHAR(50),
    @Cod_Barras  VARCHAR(50),
    @Marca       VARCHAR(50),
    @Modelo      VARCHAR(50),
    @Activo      BIT
)
AS
    SET NOCOUNT ON;
BEGIN

    DECLARE @DataHora_Sistema Datetime
    SET @DataHora_Sistema = GETDATE()
    /* A variavel @DataHora_Sistema fica com a data e hora actual*/
    DECLARE @CONT INT

    IF @Nome = ''
        return -1

    IF (SELECT COUNT(Nome) FROM v_Equipamentos_Nao_Apagados WHERE Nome = @Nome and Id != @Id ) != 0
    IF (SELECT COUNT(Cod_Barras) FROM v_Equipamentos_Nao_Apagados WHERE Cod_Barras=@Cod_Barras and @Cod_Barras != '' and Id != @Id ) != 0
    BEGIN
        RETURN -4
    end

    IF (SELECT COUNT(Nome) FROM v_Equipamentos_Nao_Apagados WHERE Nome = @Nome and Id != @Id ) = 0
    /* Efectua a contagem para verificar se na 'VIEW' existem nomes iguais para um determinado ID de Equipamento*/
    BEGIN
        IF (SELECT COUNT(Cod_Barras) FROM v_Equipamentos_Nao_Apagados WHERE Cod_Barras=@Cod_Barras and @Cod_Barras != '' and Id != @Id ) = 0
        /* Efectua a contagem para verificar se na 'VIEW' existem Codigos Barras repetidos */
        BEGIN
            /*Caso existam variaveis repetidas, retorna ERRO, caso contrario faz o Update á Tabela Equipamento*/

            SELECT @CONT = Cnt_Mod from Equipamento WHERE ID = @Id
            /*É iniciado um pequeno contador, que efectua a contagem de Update realizados*/

            UPDATE Equipamento SET
                Nome = @Nome,
                Localizacao = @Localizacao,
                Cod_Barras = @Cod_Barras,
                Marca = @Marca,
                Modelo = @Modelo,
                Data_Registo=@DataHora_Sistema,
                Activo = @Activo,
                Apagado = 0,
                Cnt_Mod = @CONT+1
            WHERE Id = @Id
```

Figura 4.12: Procedimento Modificar na tabela de Equipamentos.

- E. **Procedimento Ler:** Este procedimento é de bastante importância, pois permite ler todos os parâmetros escolhidos da tabela. Para modificar campos específicos na tabela, importa saber como estão esses campos preenchidos e com que tipos de dados. Aqui entra o procedimento Ler, retornando os elementos de um determinado *query* SQL, possibilitando assim o acesso aos dados anteriores e uma possível modificação, ou simplesmente visualização dos dados no momento.

Estes são procedimentos armazenados de extrema importância para a interação com o operador permitindo toda a dinâmica de introdução de dados. Cada um deles individualmente e para cada tabela em específico, podem existir instruções e lógica diferente, podendo interagir com outras tabelas e despoletar ações diferentes, dependendo tudo da mecânica dos processos associados ao tipo dados a movimentar.

Existem outros procedimentos independentes e relativos a tabelas específicas, que permitem o melhor funcionamento e dinâmica do sistema. O objetivo é criar os procedimentos necessários para todas as ações que se possam realizar, criando uma dinâmica inteligente e perceptível ao operador. De seguida passa-se a descrever alguns desses procedimentos e a que tabelas se referem.

F. *TagsHistorian*

O procedimento associado a esta tabela é único e permite a atualização dos valores dos contadores de máquina. Este procedimento encontra-se dentro de um SQL job, sendo executado num certo período de tempo. As instruções permitem a leitura ao *Historian* dos valores dos contadores no momento, possibilitando assim a atualização dos mesmos na tabela da base de dados. Deve ser referido que a mecânica de incremento dos contadores é feita pela ferramenta *Historian*.

G. Contadores

A tabela Contadores não é preenchida pelo operador, mas sim automaticamente aquando da inserção ou modificação de uma *schedule*. A cada máquina ou equipamento existe um contador associado, de horas de funcionamento ou de relógio e quando se cria uma *schedule* relativa a um determinado equipamento, automaticamente é criada na tabela Contadores, os respetivos parâmetros com o número da *schedule* e contador correspondente, sabendo qual a hora de início, valor inicial e atual do contador. Todas estas instruções são executadas dentro de *stored procedures* da tabela *Schedule* e não em procedimentos independentes, pois são as ações desta mesma tabela que facilitam o dinamismo e lógica existente nos contadores.

Existem ainda nesta tabela dois *stored procedures* independentes que executam ações distintas. Um desses procedimentos está diretamente interligado com a tabela *TagsHistorian* e com o procedimento da mesma.

Nesta tabela de Contadores é necessário atualizar sempre o valor desse mesmo contador, de modo a ser possível visualizar o número de horas de funcionamento da respetiva máquina. Para este processo ser realizado de forma correta, o procedimento deve ler o valor que está na tabela *TagsHistorian* e atualizar o contador respetivo com esse valor, tendo assim sempre o valor atual correto e atualizado.

Outro procedimento desta tabela é o cálculo da data de previsão para a *schedule* associada ao contador em causa. Este cálculo entra com a data de início da *schedule* e com o número de horas

em que esta estará em aviso, ou seja, num estado em que deve ser executada e transformada em ordem de manutenção. Assim o operador tem acesso à data de previsão de determinada tarefa agendada, assim como o número de horas que faltam para entrar num estado de pré-aviso ou aviso.

H. Peças

Relativamente à tabela Peças, além dos procedimentos anteriormente descritos que possibilitam a introdução, modificação ou remoção de novas peças, era necessário criar um mecanismo de introdução de peças que não o manual. Com esse objetivo, foi criado um procedimento que permite que se importe um ficheiro .csv com todo o material que se deseje introduzir, ou seja, coloca-se todo o material a introduzir para a base de dados nesse ficheiro, grava-se e basta importar. O procedimento encarrega-se de evitar que existam elementos com código e nomes repetidos, filtrando e importando o material corretamente. Este procedimento é algo complexo pois envolve tabelas temporárias que são usadas especificamente para evitar inserção de material repetido, utilizando a comparação como elemento fundamental.

Depois do material introduzido, pode-se realizar qualquer operação. Caso seja necessário introduzir mais um grande número de peças, basta exportar os dados da tabela SQL, sendo estes gravados no mesmo ficheiro .csv, com os parâmetros no momento da exportação. Este procedimento é efetuado por um *stored procedure Export Data*, que possibilita a exportação dos dados requeridos diretamente para o ficheiro .csv com a formatação correta e respetivos nomes das colunas. De seguida basta abrir o ficheiro realizar as alterações que se pretenda e importar novamente para a base de dados. Uma enorme vantagem neste processo é a possibilidade de acrescentar material à lista, fazer o *import* para o sistema, sendo que o procedimento encarrega-se de apenas importar os elementos novos. Este facto permite que se possa criar uma nova lista de material e ao inserir, as instruções limitam-se a importar o novo material inserido, ficando o restante material sempre disponível.

I. Ordens de Manutenção

Nesta tabela também existem determinados procedimentos específicos que melhoram o mecanismo da base de dados em termos de dinâmica de tarefas. Além dos procedimentos gerais, é possível também cancelar ou fechar uma ordem de manutenção. Ao criar uma ordem, esta pode estar diretamente ligada a uma tarefa agendada, ou seja, uma *schedule*, ou uma ação de manutenção espontânea.

O *stored procedure* Cancelar funciona de modo diferente nos dois casos, pois ao cancelar uma ordem de manutenção oriunda de uma tarefa agendada, a ordem de manutenção é cancelada imediatamente e a *schedule* fica no estado que se encontrava no momento em que foi despoletada a sua ordem. Isto quer dizer que, se a *schedule* se encontrava em aviso no momento em que foi criada a sua tarefa de manutenção, após o seu cancelamento ela irá voltar para esse

estado, até que nova ordem seja executada. Uma ordem de manutenção independente que seja cancelada fica inativa e só se realizará quando se criar uma nova com as mesmas ações de manutenção.

Fechar uma ordem de manutenção significa que está concluída, logo, no caso de estar associada a uma tarefa em agendamento, a ordem é fechada e a *schedule* é reposta ao seu estado inicial, ficando num estado de agenda à espera de ser executada. Uma ordem de manutenção independente é concluída com sucesso e fechada, apresentando os parâmetros finais de hora de início e fim da mesma, assim como a sua duração. Na Figura 4.13 mostra-se o exemplo do código criado para fechar uma ordem de manutenção.

```
ALTER PROCEDURE [dbo].[OrdensManutencaoCancel]
(
    @Id                INT,
    @Id_Schedule       INT
)
AS
SET NOCOUNT ON;
BEGIN
    DECLARE @DataHora_Sistema Datetime
    SET @DataHora_Sistema = GETDATE()
    DECLARE @OBSE nvarchar(1050)
    DECLARE @CONT INT
    DECLARE @PA INT
    DECLARE @TR INT

    DECLARE @Horas INT
    SET @Horas = (SELECT t1.V_Actual FROM Contador t1 INNER JOIN Schedule t2 ON t1.IdSchedule=t2.Id where t1.IdSchedule=@Id_Schedule)

    IF (SELECT COUNT(Id_Schedule) FROM v_OrdensManutencao_Nao_Apagadas WHERE Id_Schedule=@Id_Schedule) !=0 AND (SELECT Estado FROM OrdensManutencao WHERE Id=@Id_Schedule) != 4
    /*Caso exista um Id_Schedule na view indicada corresponde ao Id passado por parametro é feito os Updates das seguintes tabelas*/
    BEGIN
        UPDATE Schedule SET
            Estado= 0 ,
            Data_Registo = @DataHora_Sistema
        WHERE Id=@Id_Schedule

        UPDATE OrdensManutencao SET
            Data_Fecho_OM= @DataHora_Sistema,
            Estado=3,
            Cnt_Horas_F= @Horas
        WHERE Id=@Id

        SET @OBSE = (select 'DESC=' + cast(Descricao as varchar(100)) + ' / EQUIPA=' + cast(Equipa as varchar(50))
            + ' / USER=' + cast(Utilizador as varchar(50))
            + ' / FS=' + cast(Factor_Servico as varchar(50))
        from v_OrdensManutencao where ID=@Id)

        Insert into ScheduleActividade(Schedule, Data, Actividade, OM, Obs)
        Values (@Id_Schedule, @DataHora_Sistema, 11 , @Id, @OBSE)

        RETURN 12
    END

    /*Caso o Id_Schedule não existe é porque o OM se refere somente a um Equipameto, Conjunto ou Subconjunto, fazendo o Update só a tabela Ordens
    IF (SELECT Estado FROM OrdensManutencao WHERE Id=@Id) != 4
    BEGIN
        UPDATE OrdensManutencao SET
            Data_Fecho_OM= @DataHora_Sistema,
            Estado= 3
        WHERE Id= @Id
        RETURN 11
    END
    END
```

Figura 4.13: *Stored Procedure* utilizado para fechar uma ordem de manutenção.

4.2. System Platform

O *System Platform* da Wonderware [17] é uma plataforma de *software* industrial muito poderosa que é constituída por uma tecnologia *Archestra* para controlo de supervisão, *Geo-SCADA* e soluções de produção, gestão e manutenção de desempenho. É sobre esta tecnologia *Archestra* que se irá basear esta aplicação SCADA de gestão e manutenção de uma indústria cerâmica. É um sistema aberto e extensível de componentes com base numa linguagem orientada a objetos distribuídos. Neste caso usou-se a ferramenta principal do sistema *Archestra* [7], o ambiente de desenvolvimento integrado *IDE (Integrated Development Environment)* [6].

O *IDE* [13] é uma ferramenta de design e desenvolvimento integrado onde é permitido configurar e implementar todos os objetos *Archestra* nos computadores de destino. Com o *IDE*, é possível importar novos tipos de objetos para a Galáxia onde se está a trabalhar, configurar instâncias desses mesmos objetos e compilar essas instância em computadores da mesma rede. Entenda-se por Galáxia o ambiente de desenvolvimento onde irá ser construída toda a aplicação. Uma grande vantagem é a possibilidade de vários utilizadores poderem trabalhar simultaneamente na mesma Galáxia, em diferentes conjuntos de objetos de diferentes *IDEs*. Na Figura 4.14 é mostrado o ambiente de desenvolvimento da tecnologia *Archestra IDE*.

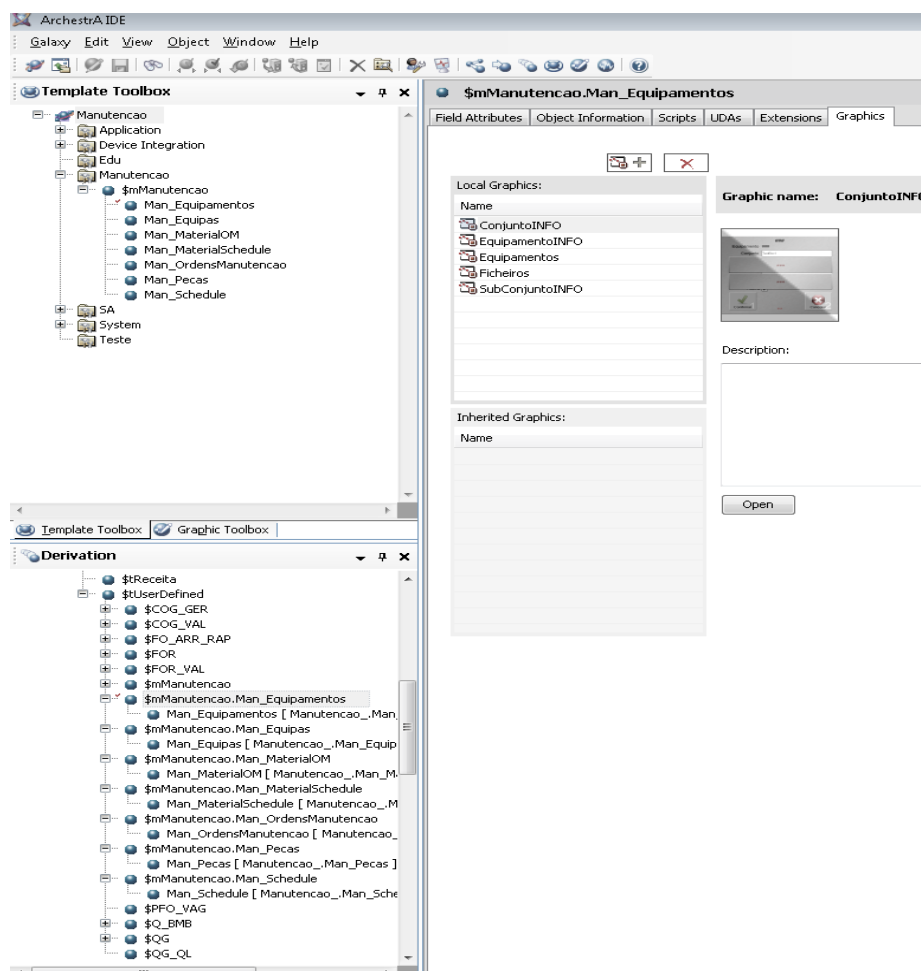


Figura 4.14: Ferramenta de design *Archestra IDE*.

Com base nesta filosofia foram criados vários objetos, caracterizados como objetos principais da base de dados de manutenção, de forma a organizar e dinamizar a programação.

Na Figura 4.15, no menu do lado esquerdo podem-se verificar os vários objetos contruídos e utilizados nesta aplicação. É sobre eles que se irá trabalhar e construir a *interface* gráfica para interligar todo o mecanismo SQL já programado. No seguimento deste capítulo serão analisados todos os objetos utilizados no sistema de gestão e manutenção, quais os processos associados e como foi pensado e criado o *interface* gráfico para cada um deles. A organização dos objetos foi pensada e elaborada tendo em conta os mecanismos já construídos em SQL *Server*, criando um objeto para cada uma das tabelas de maior importância, sendo que dentro do mesmo, são englobados todos os processos e *interface* gráfica associados a ele. Isto permite uma análise mais facilitada ao integrador e possíveis desenvolvimentos no futuro.

A. Manutenção

No objeto Manutenção encontra-se configurada a conexão ao SQL *Server*, sendo que todos os outros objetos utilizam esta conexão. Encontra-se também configurado um contador de horas utilizado em *schedules* agendadas para equipamentos que não utilizem contador de funcionamento de horas de máquina.

Esta aplicação está traduzida para Espanhol, e por esse motivo foram criadas variáveis do tipo *array* que armazenam os vários *query* SQL necessários do sistema, estando esses *query* traduzidos. Quando o operador mudar a linguagem na supervisão, automaticamente serão executadas as instruções SQL segundo a linguagem definida e as colunas das tabelas SQL traduzidas. Todas as traduções efetuadas neste sistema de gestão e manutenção estão presentes neste objeto, estando algumas delas visíveis na Figura 4.15.

Como se pode verificar também na Figura 4.15, existem quatro *scripts* neste objeto, *scripts* estes chamados, *ConnectionString*, Contador Horas Fixo, *JobsExecute* e *Language*. Como referido anteriormente, as instruções que permitem criar a ligação ao SQL *Server* estão descritas no *script* *ConnectionString*, sendo que em todos os objetos da restante aplicação, basta usar esta ligação e a conexão ao SQL é realizada com sucesso. O *script* *JobsExecute* engloba alguns procedimentos que são despoletados periodicamente pelo sistema como por exemplo, a atualização do valor dos contadores. O contador de horas de relógio encontra-se criado do *script* respetivo e utilizado quando o utilizador pretende uma contagem de horas normal. Por fim e como se pode ver na figura 4.15, as traduções encontra-se em *Language*, sendo que na imagem são mostrados todos os *query* em Português apenas. Caso na supervisão o utilizador mude a linguagem, o sistema sabe que o parâmetro da linguagem mudou, passando a executar os mesmos *query*, mas criados na linguagem pedida.

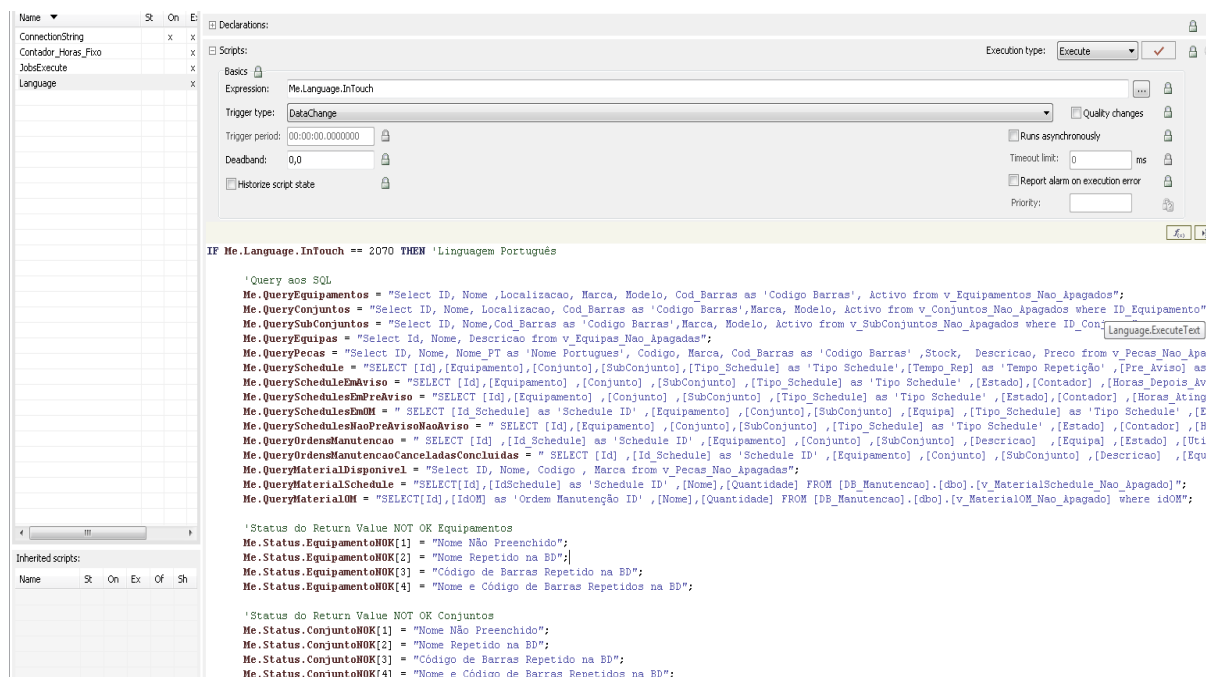


Figura 4.15: *Script* de tradução da aplicação usado no objeto Manutenção.

B. Equipamentos

O objeto Equipamentos armazena toda a informação relativa aos equipamentos, conjuntos e subconjuntos e o primeiro passo foi criar todas as variáveis de cada uma dessas tabelas SQL configurando-as corretamente. Um passo importante é fazer a ligação entre o *System Platform* e o *SQL Server* e com o objetivo de dinamizar essa ligação, usando código em linguagem *.NET*, foram criados vários *scripts* para cada tipo de *stored procedures* de ações, descritos no subcapítulo 4.1. Estes *scripts* correm assincronamente e utilizam a conexão SQL configurada no objeto manutenção, sendo que o seu objetivo é executar o respetivo procedimento SQL. Existe código *.NET* para adicionar, modificar, remover, ler e apagar equipamentos, sendo estes *scripts* configurados com as variáveis criadas em *IDE*, chamadas *UDAs*. [12,19] Assim quando se pretende realizar uma simples ação sobre um equipamento, basta chamar o respetivo *script*, que automaticamente o *stored procedure* a ele referente é executado. Na Figura 4.16 é mostrado o código criado em *.NET* para realizar o procedimento Adicionar em SQL.

Grande parte das saídas necessita de uma conjugação específica de fatores para que sejam ativadas, esses fatores dependem ou de temporizadores/contadores internos ou de variáveis de entrada da função.

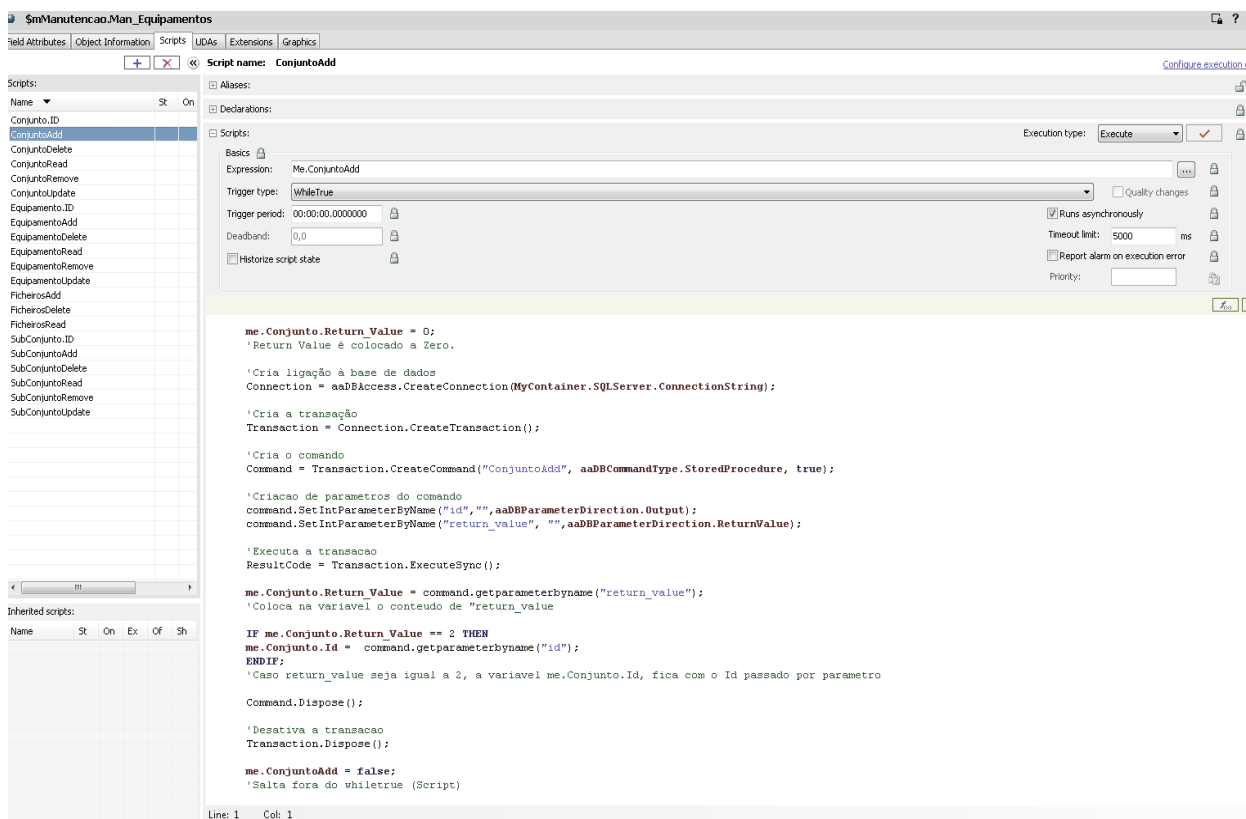


Figura 4.16: *Script* .NET representando o procedimento Adicionar.

Na Figura 4.16 é possível visualizar no menu esquerdo todos os *scripts* criados para o objeto Equipamentos, assim como o exemplo do procedimento Adicionar, mostrando a respetiva conexão ao SQL e a criação do comando que permite despoletar o procedimento de adicionar equipamento.

A interação gráfica e visual será sempre o passo final numa aplicação SCADA e é nessa fase que tudo o que está para trás, deve estar bem mecanizado e funcional, para não correr riscos de mau funcionamento. A parte gráfica é bastante importante num ambiente de supervisão porque é ela que dá a cara por toda a aplicação. Deve ser apelativa, dinâmica, simples e intuitiva, e foi com esse objetivo que se construíram todos os símbolos gráficos deste objeto e dos outros.

Na Figura 4.17 apresenta-se a janela principal do objeto Equipamentos, onde será mostrada toda a informação necessária e importante sobre os mesmos. Como se pode verificar utilizou-se o *control SQL datagrid* [10] para visualizar a informação vinda do SQL, existindo um *datagrid* individual para equipamentos, conjuntos e subconjuntos. O funcionamento da janela está pensado de uma forma dinamizadora pois ao selecionar quaisquer equipamento, no *datagrid* dos conjuntos, só irão aparecer os que pertencerem ao respetivo equipamento. O mesmo se passa para os subconjuntos, mas como é óbvio em relação aos conjuntos. Por exemplo se no *datagrid* no Equipamentos selecionamos o Forno, no *datagrid* dos Conjuntos só irão ser visíveis os conjuntos que pertencem ao Forno. Todas estas funcionalidades de visualização de dados e mecanismos de segurança, só são permitidos porque dentro deste

símbolo gráfico foram criados vários *scripts* que permitem estas funcionalidades, o que torna a janela o mais intuitiva possível.

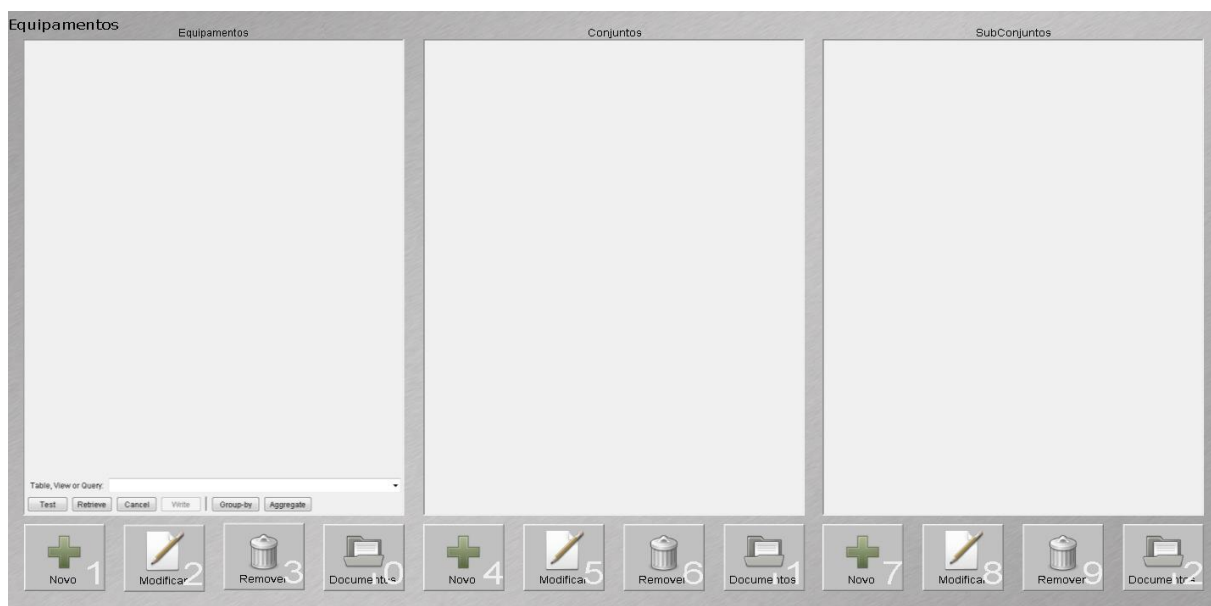


Figura 4.17: Janela de Equipamentos do objeto com o mesmo nome.

Como se pode verificar, todos os botões de ações estão numerados, sendo essa numeração só visível em desenvolvimento. Todos os botões da aplicação são símbolos gráficos gerais e não exclusivos de uma única janela. Neste símbolo gráfico, a cada botão está associado código .NET diferente para a respetiva função, sempre que existir uma mudança no valor do botão o script é despoletado e corre as instruções associados ao número referente. Qualquer botão que seja premido é chamado um *popup* de introdução de dados.

Para os equipamentos, conjuntos e subconjuntos existe um *popup* diferente, mas todos eles funcionam de forma semelhante. Sempre que um botão é premido uma janela de *popup* é aberta, nessa janela podem ser configurados os parâmetros do equipamento, como nome, localização, marca, entre outros. De seguida o operador deve dizer se o equipamento se encontra em funcionamento, ou seja, ativo ou não, sendo que por *default*, o equipamento é colocado em funcionamento aquando da sua introdução. Com os dados corretamente configurados, deve confirmar-se a operação, ou cancelar. Caso a operação seja cancelada, nenhuma alteração efetuada terá efeito. Estes símbolos gráficos de introdução de dados estão a funcionar com mecanismos de segurança, para evitar que sejam introduzidos equipamentos com nomes ou código de barras repetidos. Na imagem da Figura 4.18 mostra-se um exemplo de um símbolo gráfico de introdução de dados.

Figura 4.18: Janela de *popup* de introdução de dados do objeto Equipamentos.

Relativamente à documentação associada aos vários tipos de equipamento, existe um botão para esse efeito. Ao pressionar o botão Documentos em qualquer equipamento, conjunto ou subconjunto, irá abrir uma nova janela (Figura 4.19) onde se poderá adicionar todo o tipo de documentação associada ao equipamento aberto. Deverão ser utilizados os botões de ações dessa nova janela para adicionar a respetiva documentação à base de dados de gestão, ficando os documentos gravados na tabela de ficheiros dos equipamentos. De seguida deve-se fazer o *Load* dos documentos, ou seja, irão ser despoletadas instruções *.NET* que importam os ficheiros associados ao respetivo equipamento diretamente da base de dados. Os documentos são apresentados numa *listbox* existente no símbolo gráfico, sendo que ao premir qualquer documento, este será aberto no *web browser* do objeto. Nesta janela existe um grupo de botões e *listbox* individual para os diferentes tipos de equipamentos, estando estes grupos sobrepostos uns com os outros. Quando na supervisão é premido o botão Documentos, nos conjuntos por exemplo, automaticamente é aberta a janela, com os botões e *listbox* respetivos. Existe ainda um botão remover que permite apagar o ficheiro ou documento aberto da base de dados, ficando este imediatamente indisponível

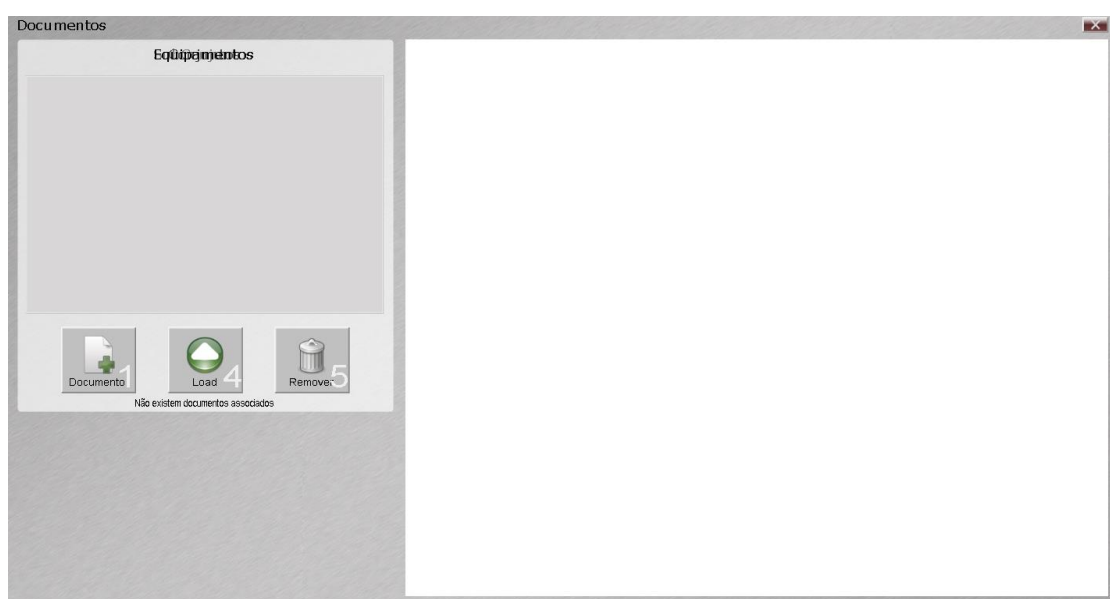


Figura 4.19: Janela de Documentação do objeto Equipamentos.

C. Equipas

O objeto Equipas é o objeto com a menor quantidade de informação da aplicação. Tal como acontece no objeto Equipamentos, foram criadas *UDAs* que representam as variáveis nas colunas em ambiente SQL, permitindo a interação com o operador e a introdução de dados corretamente. Os procedimentos gerais utilizados em SQL, são também chamados através de *scripts* .NET, havendo um por cada ação a executar.

A janela principal apresenta um único *datagrid* com a informação relativa a todas as equipas presentes na base de dados. É possível adicionar, modificar ou remover uma equipa. Mecanismos de segurança também estão presentes neste símbolo gráfico, sendo que as equipas não podem ter nomes iguais.

O *popup* existente é semelhante ao do objeto anterior, configurando-se apenas o nome da equipa e a descrição.

D. Peças

Este objeto apresenta algumas diferenças relativamente aos anteriores. Como primeira prioridade, e tal como acontece sempre nos outros objetos deve-se definir as variáveis para interagir com o SQL, estas *UDAs* devem cumprir os mesmos parâmetros que as variáveis nas colunas da tabela Peças no SQL Server. Os procedimentos são os normais, podendo o operador adicionar, modificar ou remover material, sendo que estes processos estão criados em *scripts* como comandos que executam o respectivo *stored procedure* no SQL.

Como novidade, este objeto inclui os procedimentos de importar e exportar material, procedimentos estes explicados no capítulo 4.1, sendo que neste passo foram criados *scripts* para despoletar os respetivos *stored procedures* no SQL. O material pode estar contido num ficheiro .csv com as respetivas colunas de acordo com a tabela criada em SQL Server, bastando depois importar esse ficheiro e toda a informação estará visível no *datagrid* na janela principal das peças. O *export* funciona em sentido reverso, exportando toda a informação presente no *datagrid* para um ficheiro .csv , separando corretamente todas as colunas.

Este processo apresenta uma vantagem enorme para o operador, porque para além de poder movimentar material na supervisão, pode utilizar esse ficheiro para adicionar grandes quantidades evitando perdas de tempo elevadas.

O símbolo gráfico principal das peças (Figura 4.20) apresenta igualmente o seu *datagrid* contendo a informação de todo o material no sistema. Existem os três botões de ações gerais, permitindo adicionar novas peças, modificar material existente ou remover determinada peça. Como pode existir uma grande quantidade de material na base de dados foi criado um mecanismo de pesquisa, permitindo assim ao operador encontrar determinada peça, usando o seu nome, código ou marca como referencia. Este processo é vantajoso quando se pretende adicionar determinado material nas *schedules* ou ordens de manutenção, evitando percorrer toda a lista.

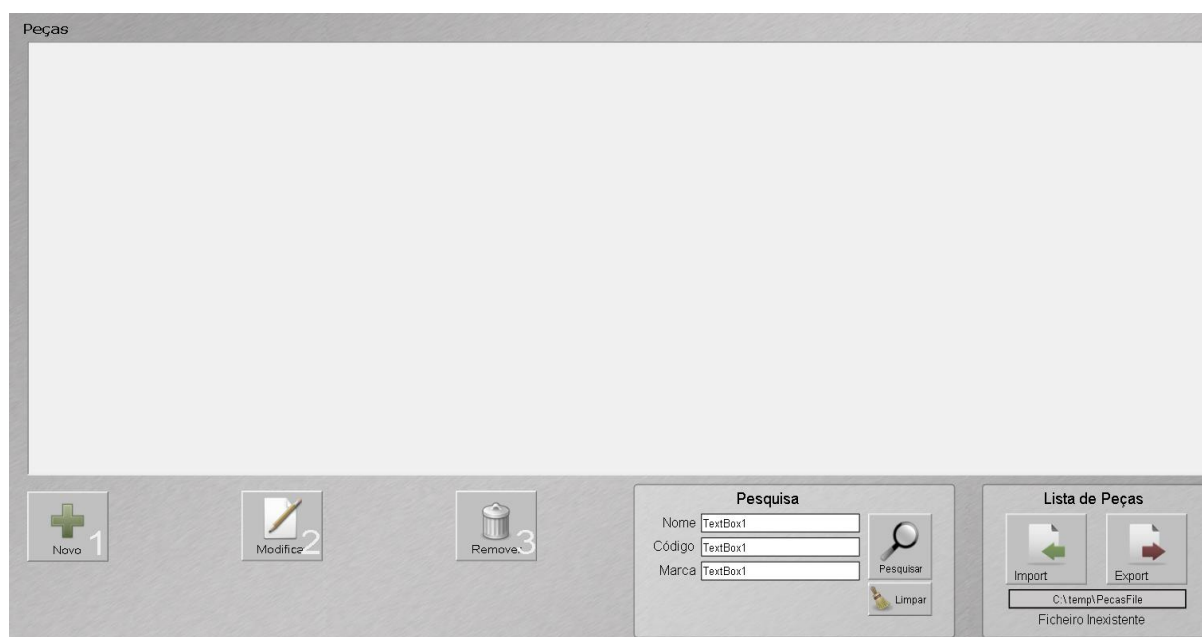


Figura 4.20: Símbolo gráfico principal do objeto Peças.

Na Figura 4.20 pode-se verificar o mecanismo de pesquisa construído com o intuito de facilitar a busca de uma determinada peça. Existem também dois botões referentes aos processos de *Import* e *Export* de material, indicando o caminho da pasta onde estará guardada a informação. Caso não exista o ficheiro na pasta indicada, quando se pretenda importar o material, surge uma mensagem de erro avisando que o ficheiro não existe.

O *popup* de peças é bem diferente dos restantes, a quantidade de informação a introduzir é maior e o nível de segurança também. Relativamente ao mecanismo de segurança, esta janela está bastante protegida evitando a introdução de material que não tenha nome introduzido ou código. Caso seja introduzido uma determinada peça que contenha um código que exista na base de dados, surge uma mensagem de erro. Sempre que se confirme a operação e surgir uma mensagem de erro, aparece também a descrição desse mesmo erro, cabendo ao operador introduzir um dado correto e prosseguir com nova confirmação. Se o botão cancelar for premido todas as alterações não têm qualquer efeito.

Todos estes mecanismos de segurança e erros que possam surgir na introdução de dados, são possíveis através de código proveniente dos procedimentos contruídos em SQL, mas também dos *scripts* criados em cada símbolo gráfico. As diferentes mensagens de erro que possam surgir são devidas a diferentes valores retornados no momento da execução dos procedimentos das peças. Esses *return values* são recebidos e com eles consegue-se saber qual o erro que aconteceu no processo. Na Figura 4.21 é mostrado um exemplo de um *script* de proteção, sendo que no lado esquerdo na imagem poder-se-ão ver todos os outros *scripts* criados para o mesmo efeito. Assim sempre que ocorrer um erro, a operação não fica concluída, sendo que o erro é descrito, para o utilizador poder corrigir e confirmar de seguida a operação.

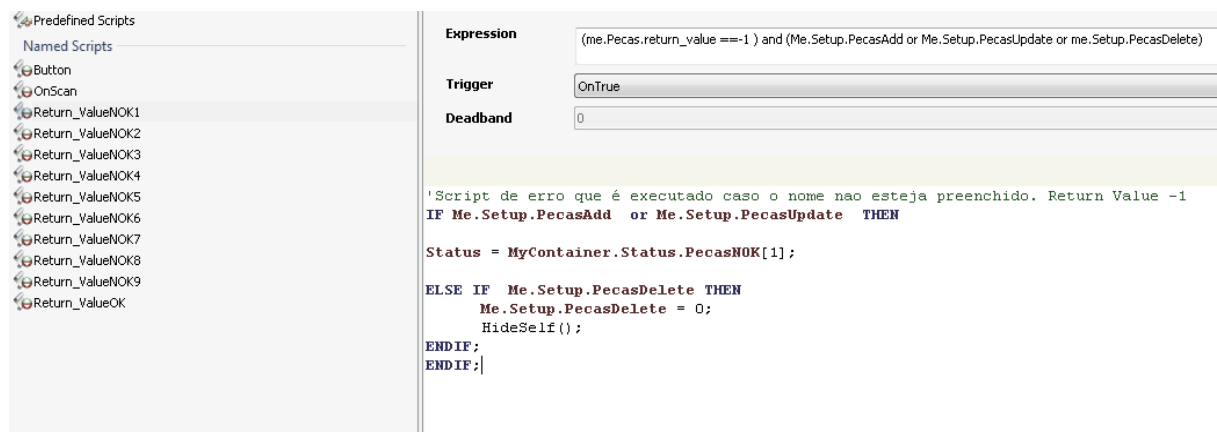


Figura 4.21: *Scripts* utilizados no símbolo gráfico de *popup* de Peças.

E. Schedules

O objeto *Schedule* será provavelmente o objeto mais importante do sistema de gestão e manutenção. As *schedules* são o centro da gestão e são estas que fazem com que essa mesma gestão seja eficiente e dinâmica. Começa-se como é habitual por criar todas as variáveis necessárias para a interação com a tabela *Schedule* do *SQL Server*, tabela esta que é central neste sistema e que vai buscar informação a muitas outras tabelas secundárias. As *UDAs* são criadas, respeitando sempre os parâmetros de configuração utilizados no *SQL Server*. De seguida criam-se os respetivos *scripts* .NET para despoletar os procedimentos responsáveis pela criação, modificação ou remoção de uma *schedule*, lembrando que estes *scripts* correm de modo assíncrono. A quantidade de informação contida nestes *scripts* é importante para uma boa dinâmica nos processos.

O símbolo gráfico principal do objeto *Schedule* é bastante parecido com símbolo utilizado nas Peças, isto é, um *datagrid* grande de modo a conter toda a informação associada a todas as *schedules*. Neste *datagrid* é chamado uns dos *Views* criados em *SQL Server* contendo informação mais precisa, utilizando o cruzamento de dados de várias tabelas. Estes mecanismos são possíveis utilizando *scripts* .NET dentro do próprio símbolo gráfico, permitindo a visualização correta dos dados, escolha de uma determinada *schedule* do *datagrid*, assim como o *refresh* sempre que exista uma alteração. Na Figura 4.22 é mostrado um exemplo de um *script* .NET utilizado para colocar visível o *datagrid* das *schedules* com toda a informação relativa às mesmas. Este código realiza a respetiva conexão ao *SQL Server*, configura todos os parâmetros necessários para a visualização do *datagrid*, correndo por fim o *query* necessário para importar os dados respetivos da base de dados.

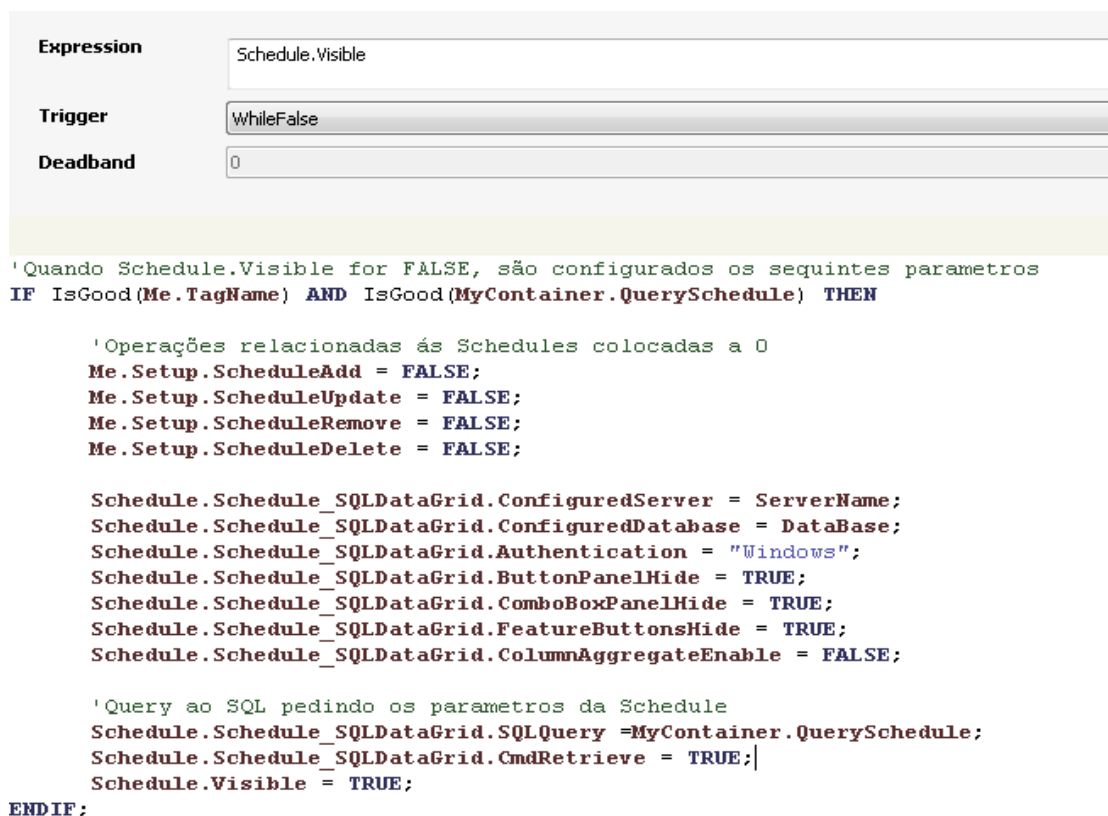


Figura 4.22: Script utilizado para mostrar os dados no *datagrid* relativos às *schedules*.

Os botões existentes no símbolo gráfico principal das *schedules* são os gerais, adicionar, modificar e remover. Estes botões ao serem premidos chamam o respetivo *popup* deste objeto.

Esta janela de *popup* é sem dúvida das mais complexas do sistema, isto porque, é um conjunto de símbolos gráficos agregados, que permitem a criação ou modificação de *schedules*, mas também a associação de material necessário para essa determinada tarefa. Vai-se analisar cada um desses símbolos gráficos e visualizar o aspeto final deste *popup*.

Relativamente ao símbolo gráfico que contém simplesmente os dados da *schedule*, este apresenta bastantes novidades relativamente aos anteriores, sendo muito mais dinâmico e interativo. As *combo box* têm um papel essencial neste símbolo, pois como existem muitos dados provenientes de outras tabelas, esses dados serão lidos para dentro desse *combo box*, tendo o operador a possibilidade de escolher qual o dado pretendido. No caso dos equipamentos e equipas, utilizaram-se mais uma vez os *scripts* .NET do próprio símbolo gráfico para permitir a leitura da informação contida nas tabelas SQL para dentro do *combo box* respetiva, assim sabe-se quais os equipamentos e equipas existentes, podendo escolher-se a que equipamentos e equipa a *schedule* se refere. Como acontece nestes últimos, o processo é exatamente igual para o tipo de *schedule* a escolher, tipo de execução e contador a associar.

O símbolo gráfico de introdução de dados na *schedule* é mostrado na Figura 4.23.

###

Equipamento

Conjunto

SubConjunto

Descrição

Equipa

Utilizador

Tipo De Schedule

Tipo De Execução

Nº Execuções

Contador

Tempo De Repetição

Pré Aviso

Factor Serviço [%] Duração Prevista [h]

Figura 4.23: Símbolo de *popup* de introdução de dados das *schedules*.

Analisando os campos de preenchimento manual destaca-se a importância do tempo de repetição, pré-aviso e fator de serviço, pois são estes campos que irão ditar a prioridade da manutenção e quando deverá ser efetuada. Relativamente ao tempo de repetição e pré-aviso, estes podem ser introduzidos em horas, dias ou meses, sendo o valor introduzido automaticamente convertido conforme a opção escolhida.

O fator de serviço determina a quantidade de horas que uma máquina está em funcionamento, sendo que este fator está diretamente relacionado com os cálculos efetuados para o tempo de pré-aviso e aviso de uma *schedule*.

Em termos de segurança o símbolo da Figura 4.23 encontra-se protegido contra alguns possíveis erros na introdução de dados, pelo que não será possível confirmar uma operação se todos os parâmetros não se encontrarem configurados corretamente. Caso não sejam introduzidas determinados parâmetros o botão de confirmação encontrar-se-á sempre desabilitado, até que tudo esteja configurado de forma correta.

Como referido no capítulo 4.1, é possível adicionar material a uma *schedule*, quer na altura da sua criação, quer em alguma futura alteração de parâmetros. Foi criado um símbolo gráfico (Figura 4.24) exatamente para esse efeito permitindo ao operador visualizar todo o material disponível e adicionar o que for pretendido. Quando um dos três botões de ações gerais no símbolo principal da *schedule* for premido, é aberto uma nova janela, e todas as ações realizadas nesse momento são referentes à *schedule* selecionada na janela principal.

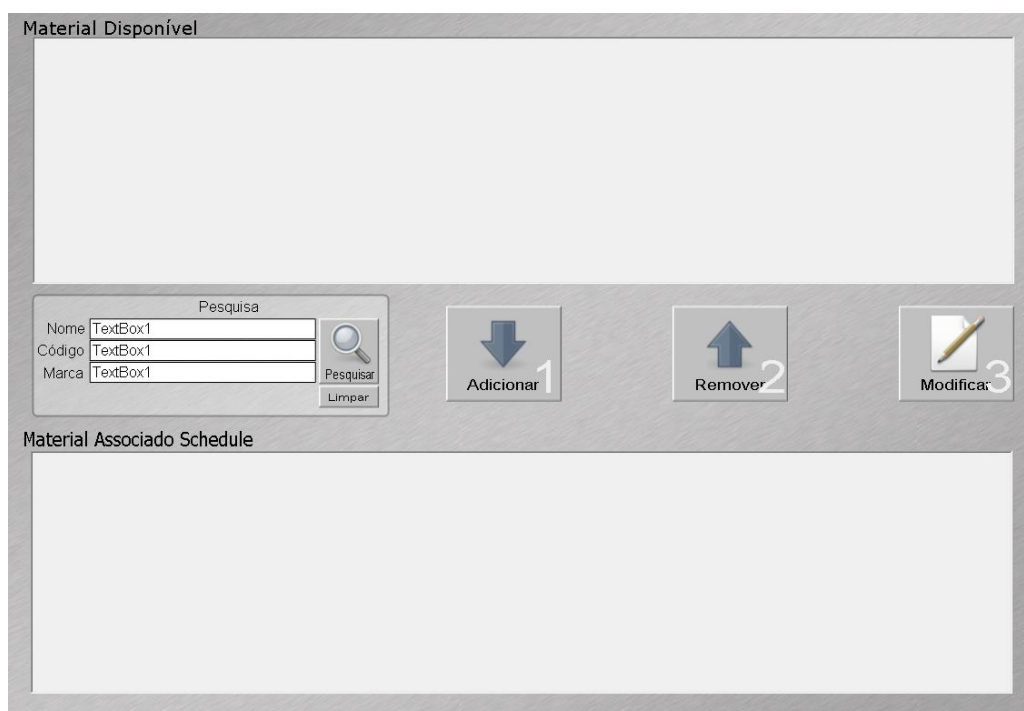


Figura 4.24: Símbolo de material relativo aos agendamentos.

Como se pode verificar existem dois *datagrid* neste símbolo, um deles referente ao material disponível e outro ao material que está associado à *schedule*. O *datagrid* para o material disponível tem exatamente a mesma funcionalidade do *datagrid* de controlo usado no objeto Peças, permitindo ter a noção de todo o material que existe em base de dados e filtrar a procura usando os mecanismos existentes. Assim o operador consegue procurar rapidamente o elemento pretendido e adicioná-lo à *schedule* respetiva.

Nesta fase da aplicação entra um novo objeto em ação, que permite efetuar as operações de adicionar, modificar ou remover material de uma *schedule* ou tarefa agendada. Este novo objeto chamado *MaterialSchedule*, tem única e exclusivamente a funcionalidade de realizar as operações descritas anteriormente. Neste objeto estão criadas as variáveis necessárias e *scripts* .NET precisos para acionar os procedimentos corretos e permitir a interação com o operador no caso de operações com material do sistema. Tal como os restantes objetos, é feita uma conexão à base de dados, sempre utilizando a conexão predefinida no objeto Manutenção, e por ação do operador são executados os *stored procedures* que armazenam as instruções necessárias que permitem a gestão de material na aplicação. Neste objeto não existem símbolos gráficos, sendo estes criados no objeto *Schedule*, chamando os procedimentos requeridos através do objeto original *MaterialSchedule*, que armazena toda essa informação. A organização da aplicação é bastante importante para existir uma maior facilidade na compreensão dos processos envolvidos por parte de um futuro integrador. Assim optou-se por criar objetos únicos para as movimentações de material, tanto nas *schedules*, como nas ordens de manutenção, sendo que qualquer operação que se pretenda efetuar deve-se chamar o procedimento pretendido armazenado nestes objetos.

Voltando ao símbolo gráfico da Figura 4.24, referente ao material, verifica-se então a existência de botões de Adicionar, Remover e Modificar material. Depois de escolhida a peça pretendida deve então utilizar-se estes botões para realizar qualquer operação. Os botões possuem *scripts* associados que chamam através do objeto *MaterialSchedule* descrito acima, os comandos necessários para realizar as operações. Depois de adicionar uma peça por exemplo, irá surgir um novo *popup* com o nome da peça que se pretende adicionar, bastando de seguida inserir a quantidade requerida e confirmar a operação. Após a confirmação da operação o material adicionado pode ser visto no *datagrid* do material associado à *schedule*, sendo permitido modificar a quantidade, se entendido.

Neste símbolo (Figura 4.24) encontram-se configuradas determinadas proteções de segurança, que não permitem ao operador adicionar qualquer material sem inserir a quantidade requisitada, nem adicionar uma peça à *schedule*, caso esta já exista. Neste último caso, surge uma mensagem de erro, dizendo que a respetiva peça já se encontra no material associado, sendo possível mudar a sua quantidade. Só é possível remover ou modificar peças se existir material associado à *schedule*, caso contrário esses botões encontram-se desabilitados.

Para finalizar a *interface* gráfica deste processo, construiu-se outro símbolo gráfico (Figura 4.25) que inclui os dois explicados anteriormente, dessa forma, dinamiza-se o processo de gestão de *schedules*, tornando-o mais eficiente para toda a manutenção.

Figura 4.25: *Popup* final de operações relativas com *schedules*.

Como existem proteções de segurança diferentes em cada um dos símbolos gráficos, os botões Confirmar e Cancelar abrangem todas essas políticas, resumindo, só é possível confirmar a operação caso sejam respeitados os parâmetros de configuração quer do símbolo que contém os dados da *schedule*, quer do símbolo que contém o material. Caso se deseje cancelar toda a operação, as alterações ficam sem efeito e a *schedule* simplesmente deixa de existir.

Como se pode ainda verificar na Figura 4.25, existe um botão de Documentos e como acontece no objeto Equipamentos, este botão tem a funcionalidade de possibilitar ao operador adicionar

todo o tipo de documentação às tarefas agendadas. Ao pressionar o botão, surge uma nova janela, onde são apresentados todos os documentos ou ficheiros associados aos equipamentos presentes nessa *schedule*, isto é, o operador ao criar uma *schedule* tem a possibilidade de visualizar no momento todos os documentos relacionados com os equipamentos a que se pretende realizar a manutenção, sendo que este facto apresenta uma vantagem enorme, pois permite que antes de se realizar qualquer tipo de manutenção sejam lidos manuais de funcionamento das máquinas e assim, configurar a *schedule* da maneira correta. No momento da criação da *schedule* é permitido também adicionar documentos à mesma, com algumas informações mais pormenorizadas e relevantes em relação à manutenção.

Ao modificar uma *schedule* existente, pode-se também ter acesso a toda a documentação inerente à mesma, ou seja, serão mostrados todos os documentos e informação relacionada com os equipamentos em causa, assim como ficheiros que estejam associados à tarefa de manutenção em agenda. Para visualizar qualquer documento, basta seleccioná-lo e este será aberto no *web browser* do símbolo. Mais uma vez, todas as operações realizadas só serão executadas com sucesso caso se confirme o processo, pois se o utilizador pressionar o botão cancelar, tudo o que se possa ter efetuado nesta janela, ficará sem efeito.

Na Figura 4.26 mostra-se o aspecto final do símbolo gráfico relativo aos documentos associados às *schedules*.

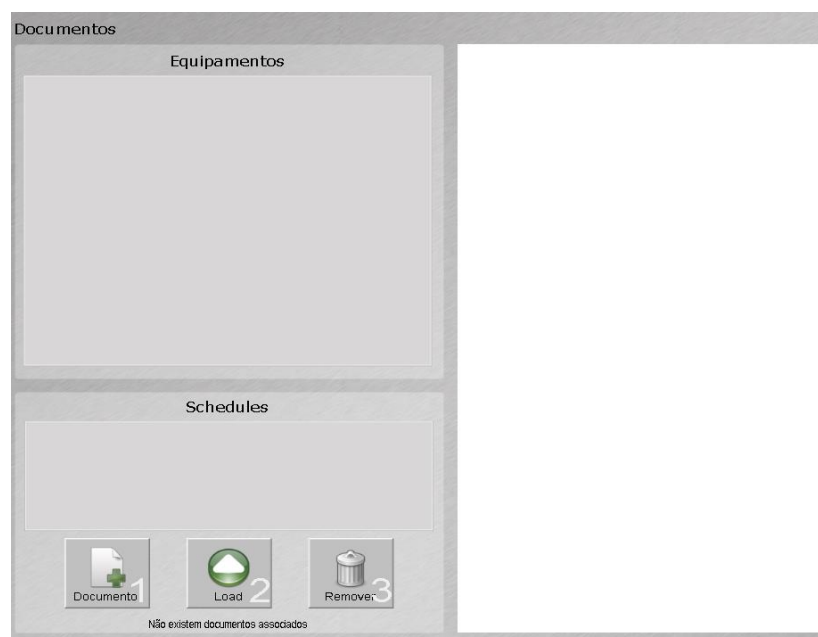


Figura 4.26: Janela de documentação do objeto *Schedule*.

F. Ordens de Manutenção

O objeto Ordens de Manutenção é o objeto com mais informação e mais complexo do sistema. Para além de estar diretamente relacionado com as *schedules* e receber informação da mesma, também interage com diferentes tabelas e diferentes dados. As ordens de manutenção podem ser originárias de uma *schedule* agendada ou de uma ação espontânea ou urgente, e assim o

objeto foi adaptado para atender aos diferentes tipos de necessidades. Em termos de variáveis declaradas o objeto é como habitual semelhante aos restantes, estando as *UDAs* criadas de acordo com os parâmetros SQL. Os *scripts* respeitam as mesmas regras dos anteriores, sendo despoletados quando o operador assim o entender. É importante referir, como foi mencionado no capítulo referente aos *stored procedures* SQL, este objeto apresenta diferentes procedimentos relativamente aos restantes, como fechar uma ordem de manutenção, cancelar ou saltar.

O símbolo gráfico (Figura 4.27) principal deste objeto é completamente diferente dos restantes, contendo uma maior quantidade de informação e um maior número de operações possíveis de realizar. Este símbolo interage diretamente com as tarefas agendadas sendo importantes relacionar estas tarefas com a ordem de manutenção que as irá executar. Como era este o objetivo pretendido, entendeu-se ser importante englobar toda essa informação numa só janela, para que o operador tivesse a noção dos vários tipos de manutenção possíveis e as diferentes prioridades.

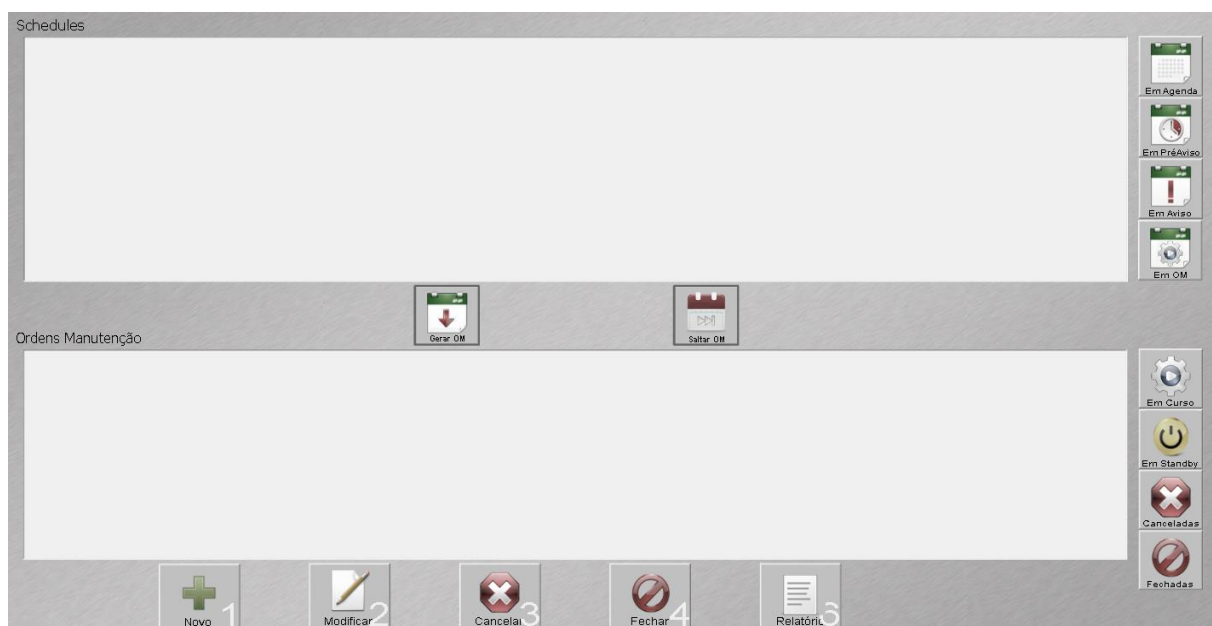


Figura 4.27: Janela principal do objeto Ordens de Manutenção.

Como se pode verificar, existe um *datagrid* relativo às *schedules*, que nada tem a ver com o *datagrid* com o mesmo nome do objeto *Schedule*, pois neste caso, não são mostradas todas as tarefas agendadas existentes e os seus respetivos parâmetros. Quando aberta esta janela, no *datagrid schedules*, é apresentada a informação das tarefas que estão em agenda, mas não se encontram com prioridade elevada, contendo assim as respetivas horas que faltam para entrar em pré-aviso e aviso.

Voltando um pouco atrás, ao ponto relacionado com os *Views* SQL, este *datagrid* está diretamente relacionado com esses mesmos *Views*, pois toda a informação mostrada é proveniente de instruções *SELECT* executadas nos diferentes *Views*. Assim como se pode visualizar, existem diferentes botões de filtros das *schedules*, sendo que cada botão representa

um *View* diferente, e cada um deles despoleta uma ação distinta, sendo assim possível ver todas as *schedules* que estão em, pré-aviso, em aviso ou mesmo, as que já se encontram em fase de execução, isto é, estejam com ordem de manutenção.

Estes *Views* contêm informação importante para a gestão e manutenção das *schedules*, pois pode saber-se quantas horas faltam para a tarefa entrar em aviso, ou quantas horas passaram desde que entrou em aviso, sabendo também sempre qual a data prevista para se realizar a manutenção.

Quando se pretende gerar uma ordem de manutenção através de uma *schedule* agendada, basta seleccionar a *schedule* pretendida no *datagrid* respetivo e seleccionar o botão Gerar OM, um novo *popup* será aberto. Relativamente a este *popup* ou símbolo gráfico (Figura 4.28), é bastante semelhante ao símbolo (Figura 4.23) que continha os dados da *schedule* no objeto anterior. Neste caso como se está a criar uma ordem de manutenção oriunda de uma tarefa agendada, ao gerar a tarefa os parâmetros da *schedule* serão automaticamente associados a essa ordem, sendo que o operador introduz apenas o utilizador e confirma a operação. Neste *popup* os equipamentos, equipa e descrição serão os mesmos que estejam definidos na *schedule* agendada.

###

Schedule

Equipamento

Conjunto

SubConjunto

Equipa

Estado

Descrição

Utilizador

Figura 4.28: Símbolo gráfico relativo a uma ordem de manutenção gerada de uma *schedule*.

Neste símbolo (Figura 4.28) as *combo box* encontram-se sempre desabilitadas, pois quando a janela é aberta são lidos nessas *combo box* todos equipamentos e equipa a que a tarefa agendada se refere, o operador não pode alterar essa informação.

Como sucede no objeto *Schedule*, é criado um símbolo gráfico apenas para as operações de movimentação de material, logo de forma análoga, existe um objeto chamado Material Ordens Manutenção, que é usado apenas para se poderem realizar estas operações de adição, remoção ou modificação de material. O objeto é semelhante ao *MaterialSchedule*, mas associado às ordens de manutenção. Também neste símbolo, existe um *datagrid* com o material disponível

de toda a base de dados, e outro *datagrid* com material associado às ordens de manutenção. Como particularidade, neste caso, quando se transforma uma *schedule* agendada numa ordem de manutenção, todo o material que lhe estiver associado passa imediatamente a estar acessível no *datagrid* de material das ordens de manutenção, sendo possível depois alterar, apagar ou adicionar mais material. Em termos de proteções continua a não ser possível adicionar material que já existe nas ordens de manutenção, sendo obrigatório então modificar a quantidade. Caso se adicione qualquer peça que exista na ordem, imediatamente surge um aviso, sugerindo que se mude então a quantidade na peça já existente. O aspeto final deste símbolo é muito parecido com o símbolo da Figura 4.24.

Por fim é criado um símbolo gráfico (Figura 4.29) contendo os dois símbolos mencionados anteriormente, como acontece no objeto *Schedule*. As políticas de segurança de ambos os símbolos encontram-se ativas e só é possível confirmar a operação, se respeitadas todas as regras e configurações. É importante referir que todos os símbolos descritos são exclusivamente relativos a ordens de manutenção oriundas de tarefas agendadas.

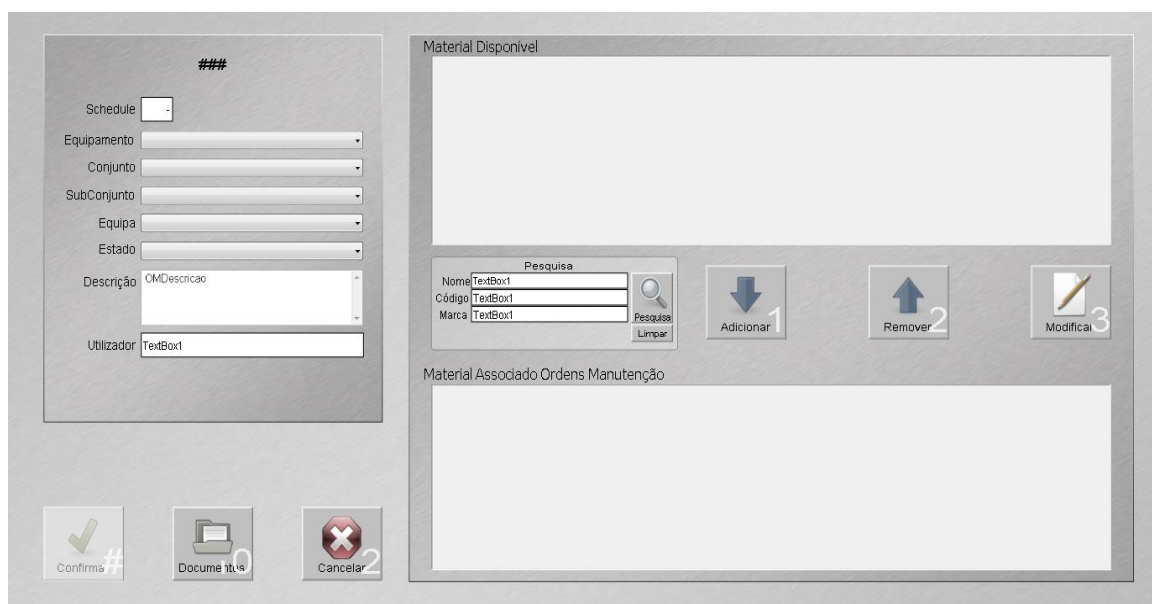


Figura 4.29: Símbolo gráfico relativo a ordens de manutenção geradas a partir de *schedules*.

Pode verificar-se que o símbolo (Figura 4.29) é bastante parecido com o símbolo utilizado no objeto *Schedule*, sendo este exclusivamente relativo a ordens de manutenção agendadas. Também esta janela possui um botão de documentos, que serve para guardar documentação relativa às ordens, mas deste assunto falar-se-á mais a frente.

Voltando à janela principal deste objeto (Figura 4.27) é possível avançar com a *schedule* para a próxima data de execução. Este facto pode acontecer quando por exemplo, a tarefa deveria ter sido executada mas ficou pendente, logo, pode efetuar-se o avanço da mesma para a próxima execução, ficando novamente em agenda. Para efetuar o processo, basta premir o botão *SaltarOM*, depois de premido, surge um *popup* no ecrã, para introdução do nome do utilizador que vai realizar a ação, e um campo de observações onde se deve descrever o porquê de saltar a execução da respetiva tarefa. Realizados estes passos é possível confirmar a operação e

avancar com a *schedule* para a próxima data de execução, recalculando as horas que restam para entrar em pré-aviso e em aviso.

No símbolo gráfico principal (Figura 4.27) deste objeto Ordens de Manutenção, pode-se visualizar todas as ordens de manutenção, em curso, standby, canceladas ou fechadas. Mais uma vez os *Views SQL* surgem aqui como uma ferramenta importante para a visualização de dados, sendo que cada botão desses representa um *View* independente, com parâmetros diferentes também, de modo a fornecer uma informação mais objetiva e precisa acerca das ordens de manutenção. É importante salientar que como dito anteriormente podem ser realizadas operações com as ordens de manutenção. Ao selecionar o botão Novo, está-se a criar uma ordem de manutenção independente e espontânea, que não está relacionada com nenhuma *schedule*, e neste caso o operador utilizando uma outra janela mas em tudo igual a da Figura 4.28, determina quais os equipamentos onde se pretende realizar a operação, por quem será realizada e qual será essa tarefa. É possível adicionar material à ordem, caso assim seja pretendido.

Ao modificar uma ordem de manutenção, só alguns parâmetros são acessíveis à alteração sendo que movimentações de material são sempre permitidas.

Quanto às ações de cancelar e fechar, caso estas se referirem a ordens de manutenção independentes, são executados os respetivos procedimentos de cancelamento ou conclusão e pode visualizar-se as datas em que ocorreram as ações, assim como as horas de duração da ordem.

Caso as ordens de manutenção sejam provenientes de *schedules*, e como referido no capítulo relativo ao *SQL Server*, ao cancelar essa ordem, a respetiva *schedule* retorna para o estado que tinha antes de ser executada a sua tarefa. Ao fechar uma ordem de manutenção, esta será concluída com êxito, e a *schedule* associada, é recalculada, voltando a estar em agenda com os mesmos parâmetros de configuração iniciais. É importante referir que ao cancelar ou fechar uma ordem, só é possível confirmar ou cancelar a operação, ou seja, não é possível realizar qualquer outro tipo de ação.

Pode-se sempre ter acesso a todas as ordens que foram canceladas ou fechadas bastando para isso selecionar o respetivo botão.

Relativamente ao botão de Documentos, da Figura 4.29 este apresenta algumas diferenças em relação ao mesmo utilizado no objeto *Schedule*. A janela que é aberta depois do botão de Documentos ser premido, serve para guardada a documentação relativa a todas as ordens de manutenção efetuadas, mas como se sabe, essas ordens podem surgir de tarefas agendadas ou de necessidade momentânea. Assim, tendo este facto em consideração, sempre que se estiver a gerar ou modificar uma ordem de manutenção já agendada, ao selecionar o botão, irá surgir nas *listbox* respetivas, a informação e documentos referentes aos equipamentos da *schedule*, assim como os documentos e ficheiros associados à tarefa em agenda. O operador tem a possibilidade então de adicionar documentação referente à ordem em causa, sendo esta guardada na base de

dados, na tabela respetiva. Toda a informação e documentos tanto dos equipamentos como da *schedule* são exclusivamente para leitura.

No caso de uma ordem independente, é só mostrada a documentação relacionada com os respetivos equipamentos escolhidos. O operador continua a ter a possibilidade de guardar qualquer documento relevante relacionado com a ordem de manutenção. Neste caso, é importante referir que não é mostrada a *listbox* de *schedules*, pois trata-se de uma ordem independente, não sendo necessária qualquer informação das mesmas. Sempre que não exista documentação associada a qualquer um dos objetos, surge um aviso no ecrã.

Como acontece no objeto *Schedule*, qualquer operação efetuada, só terá efeito caso seja confirmado o processo. Logo, deve-se realçar que toda a movimentação que existir com adição de material ou documentos por exemplo, só será processada com sucesso depois de confirmada a operação. Com o cancelamento de todo o processo, as alterações ficam sem efeito, perdendo o operador todas as modificações efetuadas.

Na Figura 4.30 mostra-se o símbolo gráfico contruído para guardar todos os documentos inerentes às ordens de manutenção. Mais uma vez *scripts* .NET dentro do próprio símbolo são de extrema importância para a leitura de todos os ficheiros relativos aos respetivos equipamentos, *schedule* e ordem neste caso, assim como permitem que exista a possibilidade de poder ler e apresentar qualquer tipo de documento, independentemente da sua extensão. São instruções existentes nestes *scripts* que permitem que sejam lidos os ficheiros guardados na base de dados, saber que tipo de extensão esses ficheiros possuem e sejam criadas pastas para armazenar temporariamente os documentos relativos a equipamentos, *schedule* e ordem de manutenção respetivamente. De seguida basta fazer um *retrieve* aos ficheiros armazenados em cada pasta e colocá-los nas *combo box*, para leitura na supervisão. Sempre que esta janela é fechada essas pastas são apagadas sendo que toda a documentação continua a ser guardada na BD, podendo ser apagada caso o operador assim o entenda.



Figura 4.30: Símbolo gráfico referentes aos documentos relacionados com ordens de manutenção.

Existe também um botão de relatório na janela principal das ordens de manutenção da Figura 4.27, relatório este criado em *Report Builder 2.0*. Ao selecionar uma ordem de manutenção e em seguida premir o botão Relatório, são mostrados todos os dados importantes relativos à ordem escolhida, e no caso deste relatório, está feito para ser impresso e o operador preencher manualmente todos os parâmetros essenciais à tarefa executada como data de início, data de fecho, técnico da operação e descrição do trabalho efetuado.

G. Reports

Neste último objeto, é onde se encontram armazenados todos os relatórios do sistema de gestão e manutenção. Estes relatórios foram criados através do *software Report Builder 2.0* e são uma ferramenta importante para análise de dados relativos a toda a manutenção. Os relatórios foram criados de uma forma dinâmica, usando instruções diretas ao *SQL Server*, para retornar os dados pedidos no momento. Foram criados diversos relatórios para atender às diversas necessidades de gestão e possibilitar uma melhor capacidade de análise e eficiência nas decisões tomadas.

Report Builder 2.0

O *Report Builder* é uma ferramenta usada para criar relatórios de base de dados. Ao contrário do *Report Wizard*, usando o *Report Builder* pode-se assumir o controlo e projetar o relatório da maneira que se entender. O relatório é gerado num documento de texto, pelo que pode ser editável.

O primeiro passo para a realização dos relatórios é criar uma *data source* que representa a conexão a base de dados com que se está a trabalhar, esta *data source* pode ser usada nos restantes relatórios.

O primeiro relatório criado foi uma listagem de equipamentos (Figura 4.31), ou seja, são apresentados todas as combinações de equipamentos existentes e se a esses equipamentos estão associadas *schedules* ou ordens de manutenção. Caso existam, é possível selecionar a *check box* relativa a cada uma delas e automaticamente salta-se para outro relatório onde são mostrados todas as tarefas agendadas ou ordens em execução, associadas aos respetivos equipamentos, dependendo da informação que se pretenda aceder.

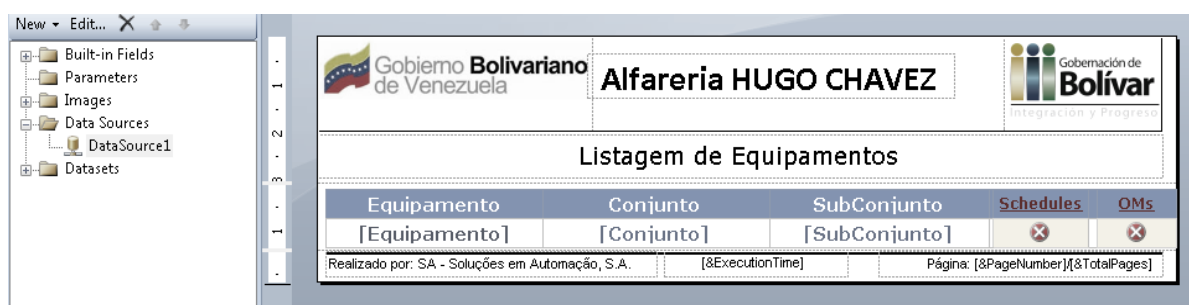


Figura 4.31: Construção do relatório relativo à listagem de equipamentos existente.

Quando o relatório é gerado, cada equipamento surge no campo respetivo, sendo que no campo de *schedules* ou ordens de manutenção, se existirem, aparece um *check* interativo que ao ser premido, permite saltar para outro relatório. Caso não existam tarefas agendadas ou ordens em algum equipamento, o campo é desabilitado, permanecendo o sinal vermelho com a cruz.

É importante referir que na criação dos relatórios, umas das principais preocupações foi a possibilidade de existir tradução para várias línguas, sendo que um dos parâmetros iniciais de todos os relatórios é a linguagem que se pretende utilizar. De acordo com a linguagem escolhida o relatório irá seguir as instruções SQL respetivas e apresentar o relatório nessa mesma linguagem. Todas as traduções relativas aos relatórios do sistema encontram-se numa tabela da base de dados SQL *Server* chamada *ReportTranslations*. Existem traduções feitas para espanhol e inglês, para além de português.

1. Schedules

Relativamente às *schedules* existem relatórios de informação por equipamento, por *schedule* individual, por estado ou uma previsão das *schedules* agendadas. É possível saltar sempre entre relatórios, tornando o processo muito mais dinâmico e os dados mais acessíveis ao operador.

Nestes relatórios existem parâmetros de entrada, o operador deverá introduzir a linguagem que pretende para os relatórios e os equipamentos associados, nos quais pretende ver as *schedules* agendadas. Assim neste relatório (Figura 4.32) tem-se acesso a todas as tarefas agendadas para o equipamento escolhido, alguma informação relevante em relação às mesmas e a possibilidade de saltar para uma *schedule* específica, bastando para isso, seleccionar o ID pretendido.

Language: pt-PT | Equipamento: Preparacion | Conjunto: Desintegrador | Sub Conjunto:

1 de 1 | 100% | Localizar | Próximo

Gobierno Bolivariano de Venezuela | **Alfareria HUGO CHAVEZ** | **Gobernación de Bolívar**

Sistema de Gestão e Manutenção
Agenda Por Equipamento

Equipamentos			
		Marca	Modelo
Equipamento	Preparacion		
Conjunto	Desintegrador	VERDES	127 C

Schedules				
Id	Tipo de Schedule	Contador	Equipa	Descrição
34	Contadores Horas Máquina	Preparación		Compruebe el aceite y el nivel de aceite del motorreductor.
35	Contadores Horas Máquina	Preparación		Engrasar el conjunto de rodamientos tipo aceite G2
36	Contadores Horas Máquina	Preparación		Lubricar sistema neumático rasquetas capacidad=0,02dm3 tipo aceite A7

Realizado por: SA - Soluções em Automação, S.A. | 24-09-2014 02:04:04 | Página: 1/1

Figura 4.32: Relatório de tarefas agendadas por equipamento.

O relatório de informação de uma *schedule* específica, tem informações pormenorizadas da *schedule*, sabendo o operador, qual a tarefa em causa, qual a operação a efetuar, quais os parâmetros associados, as respetivas horas que possam faltar para a tarefa ser executada e qual o estado que se encontra atualmente.

Relativamente às tarefas agendadas por estado, existe um relatório onde pode visualizar-se quantas *schedules* estão agendadas para os diferentes estados, assim como quantas estão em ordem de manutenção. Para consultar as *schedules* basta carregar no icon do +, selecionando também o mesmo símbolo para ver mais informações acerca das mesmas.


O relatório de previsão das *schedules* agendadas mostra todas as *schedules* que se encontram em agenda, isto é, não se encontram com prioridade elevada. São mostradas também as tarefas agendadas que se encontram em pré-aviso para datas predefinidas, sendo possível, visualizar quais entram em aviso dentro da próxima semana, duas semanas, ou um mês por exemplo. É importante salientar que tanto nos relatórios por estado como, relatórios de previsão, é sempre possível visualizar informação pormenorizada das *schedules* em causa, selecionando o ID respetivo.

Na Figura 4.33 mostra-se o exemplo de um relatório relativo a uma tarefa agendada.

Página Inicial > Manutenção > AgendaPorSchedule

Language ID

1 de 1 100% Localizar | Próximo

 **Gobierno Bolivariano de Venezuela** **Alfareria HUGO CHAVEZ**  **Gobernación de Bolívar**

Sistema de Gestão e Manutenção
Informação De Schedule

Equipamentos			
Equipamento	Conjunto	SubConjunto	
Preparacion			
Schedule ID	16		
Tipo de Schedule	Contadores Horas Máquina		
Equipa			
Contador	Preparación		
Descrição	En todos los motoreductores de las cintas transportadoras. Cambiar el aceite mineral; substituya la grasa para rodamientos y el réten del eje de salida (no montarlo de nuevo sobre la misma huella).		
Tempo de Repetição (h)	Pré Aviso (h)	Duração Prevista (h)	Factor de Serviço (%)
11500	720	0	100
Estado da Schedule			
Em Funcionamento			
Horas Para Atingir Pré Aviso	25771		
Horas Para Atingir Aviso	26491		
Horas Depois do Aviso			
Data Registo	02-09-2014 03:53:48		
Utilizador	Carlos		

Realizado por: SA - Soluções em Automação, S.A. 24-09-2014 02:10:55 Página: 1/1

Figura 4.33: Relatório relativo a uma *schedule* individual com toda a informação associada.

2. Ordens de Manutenção

Como acontece nas tarefas agendadas, para as ordens de manutenção executadas existem relatórios por equipamento, por ordem ou por estado.

O funcionamento dos mesmos é semelhante aos anteriores, mudando o tipo de informação, ou seja, no caso do relatório por equipamento, são mostradas todas as ordens de manutenção associadas a esse equipamento, sendo possível saltar para a ordem em causa e verificar em pormenor a informação que lhe está associada. É permitido também saltar para a *schedule*, caso essa ordem seja proveniente de uma *schedule* agendada.

No relatório por ordem de manutenção, tem-se acesso a mais informação, sabendo em pormenor qual a operação efetuada na manutenção, o material utilizado na mesma, a duração e qual o estado em que se encontra atualmente.

Por fim, no relatório relativo aos diferentes estados de uma manutenção, tem-se acesso a todas as ordens, em agenda, fechadas ou canceladas e a toda a informação que a elas corresponde. Este relatório encontra-se filtrado com parâmetros de data de início e data de fim, sendo só possível verificar todas as ordens de manutenção canceladas ou concluídas das datas escolhidas.

Na Figura 4.34 pode visualizar-se um exemplo deste ultimo relatório descrito, podendo o operador verificar quais as ordens de manutenção existentes nas datas escolhidas e quais os estados em que se encontram.

Sistema de Gestão e Manutenção					
Informação De Ordens De Manutenção					
☐ Ordens De Manutenção Em Curso					7
Id OM	IdSchedule	Equipamento	Conjunto	SubConjunto	Descrição
☐ 7	46	Preparacion	Cinta Transportador a 06		Limpiar los dientes de las palas espalhadoras
☐ 8	45	Preparacion			En todos los raspadores de las cintas transportadoras. Verificar y ajustar la posicion de la banda raspadora.
☐ 11	41	Preparacion	Amasadora		Engrasar los rodamientos tipo aceite G2
☐ 21	0	Preparacion	Laminador		
☐ 23	79	Forno			Limpeza Geral do forno
☐ 57	39	Preparacion	Laminador		Lubricar sistema neumático rasquetas capacidad=0,02dm3 tipo aceite A7
☐ 74	32	Preparacion	Alimentador Lineal Escamas Metalicas Eje Rompedor		En los motoreductores Bonfiglioli. Verificar retenes externos y juntas; Control esmerado del desgaste o posible envejecimiento de los retenes exteriores con posibilidad de substitución de los compon
☐ Ordens De Manutenção Em StandBy					0
Ordens De Manutenção Entre 05-08-2014 > 28-09-2014					
☐ Ordens De Manutenção Canceladas					0
☐ Ordens De Manutenção Concluídas					0
Realizado por: SÁ - Soluções em Automação, S.A. 24-09-2014 02:39:36 Página: 1/1					

Figura 4.34: Relatório apresentando todas as ordens de manutenção e os seus estados relativamente às datas escolhidas.

H. Menu

Mesmo não sendo nenhum objeto pois trata-se apenas de um símbolo gráfico, foi criado um menu representando cada um dos objetos do sistema (Figura 4.35), com diferentes botões que consequentemente abrirão as respectivas janelas principais de cada objeto. O mecanismo criado por trás da seleção de cada botão foi conseguido através de *scripts* .NET, estando a cada botão atribuído um número, que seguidamente se encontra relacionado com a janela respectiva.



Figura 4.35: Menu relativo ao sistema de gestão e manutenção da aplicação

4.3. InTouch HMI

Chega-se à última peça desta aplicação, o *InTouch*. É através desta poderosa solução *Wonderware* que se vai construir o aspeto final da supervisão, aproveitando todas as ferramentas e vantagens que a solução apresenta. O objetivo era contruir uma aplicação robusta, bastante eficiente mas também intuitiva ao utilizador de forma a permitir uma facilidade na gestão das operação de controlo e manutenção [4] .

Como dito anteriormente no capítulo 3.1 referente ao *InTouch*, a monitorização das aplicações, ocorre por meio de símbolos gráficos com indicações dinâmicas do estado dos equipamentos e grandezas analisadas, e tendo esse objetivo em mente, optou-se por construir um menu com diferentes janelas representando cada um dos objetos produzidos em *System Platform* [18].

Foram construídas as várias janelas com os nomes adjacentes aos objetos criados anteriormente, sendo que é possível saltar entre as mesmas, abrindo a pretendida (Figura 4.36). Dentro destas janelas *InTouch*, será colocado o símbolo gráfico principal de cada objeto, associado claro à janela correspondente, desta forma, ao selecionar-se um botão no menu principal, garante-se que se associa o símbolo correto à janela a ser aberta.

A aplicação está criada com dois níveis de segurança, ambos com *username* e *password*, os níveis estão definidos como Administrador e Operador.

De referir que a nível de linguagem, é no *InTouch* que se estabelece em qual linguagem se pretende visualizar a aplicação, sendo que esse parâmetro vai comunicar diretamente com o objeto Manutenção, de maneira a apresentar a tradução correta.

Entrando em *Runtime*, pode-se visualizar todo o aspeto final da aplicação e a sua interação com o operador. De seguida será dada uma breve explicação de como pode ser usado este módulo de gestão e manutenção aplicado à indústria cerâmica.

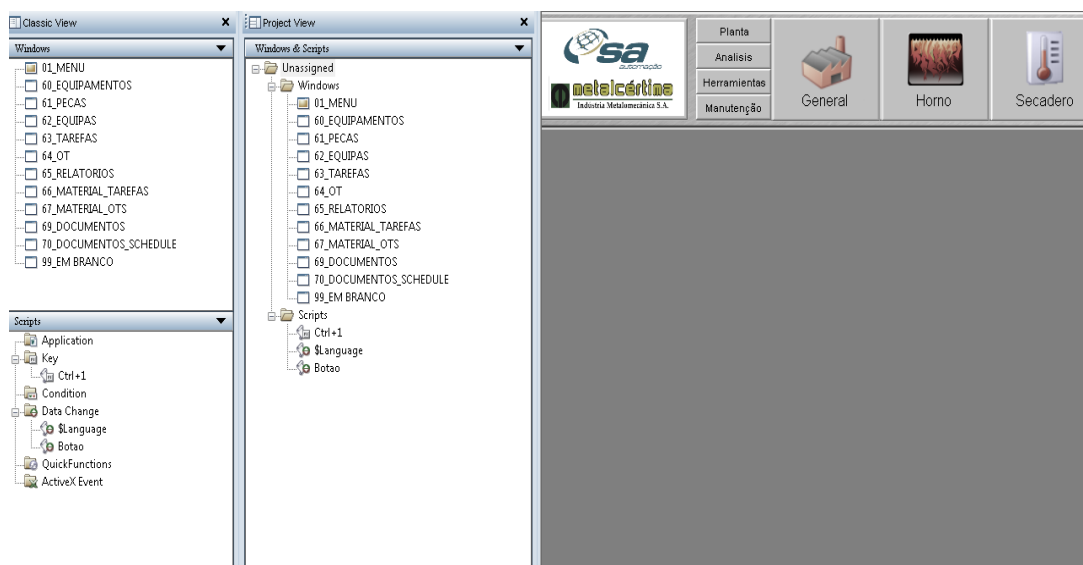


Figura 4.36: Ambiente gráfico *InTouch* com as janelas da aplicação de gestão e manutenção definidas.

A. Equipamentos

Ao entrar na aplicação de manutenção, tem-se acesso ao menu inicial no qual se pode escolher a janela a ser aberta. No caso apresenta-se a janela de Equipamentos (Figura 4.37), onde se pode consultar e editar equipamentos, conjuntos e subconjuntos do sistema. Como referido anteriormente, tem-se os vários botões de ações para os diferentes tipos de equipamento. Ao remover quaisquer equipamentos, todos os que lhes estiverem associados serão igualmente removidos.

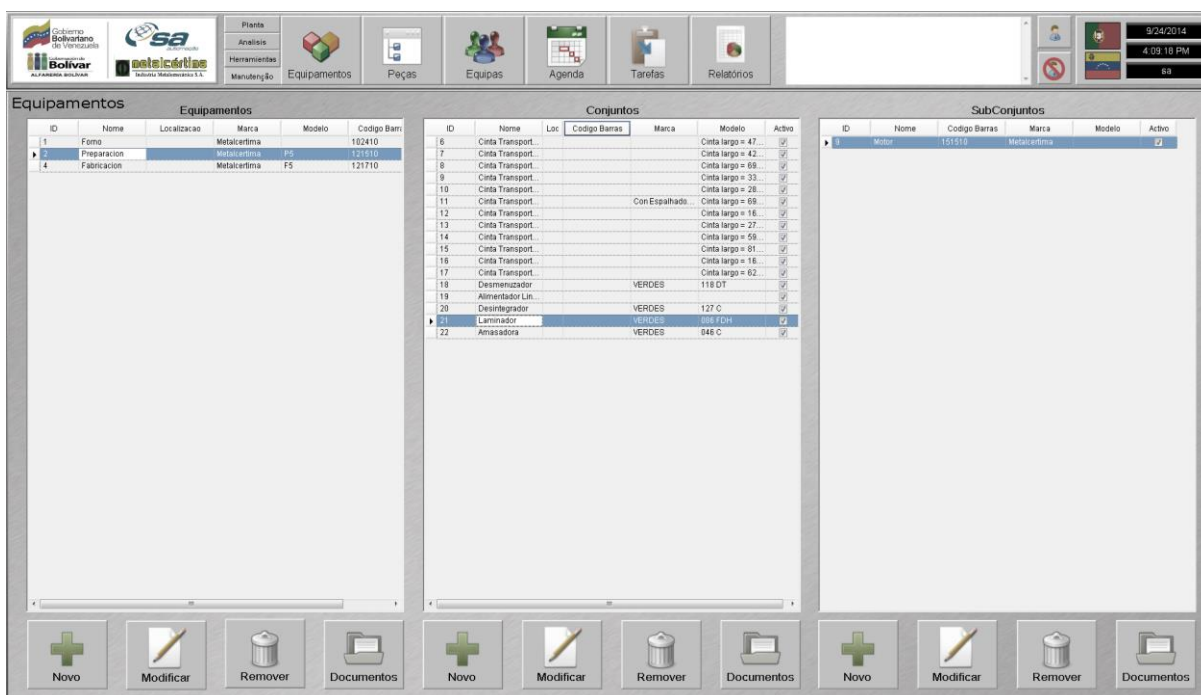


Figura 4.37. Janela de Equipamentos relativa à supervisão em *InTouch*.

Para por exemplo, modificar um equipamento no sistema, deve-se escolher o equipamento, seleccionar o botão Modificar e na nova janela alterar todos os campos pretendidos, indicando se o equipamento está ativo ou não. Para finalizar a modificação do equipamento, basta seleccionar Confirmar como pode ser visível na Figura 4.38.

A janela intitulada "Modificar Conjunto" apresenta um formulário para alterar os dados de um equipamento. O formulário contém os seguintes campos:

- Equipamento:** Preparacion
- Conjunto:** Laminador
- Localização:** LocB
- Código Barras:** 151019
- Marca:** Verdes
- Modelo:** 086 FDH
- Activo:** ☒

Na base do formulário, há dois botões: "Confirmar" (com um ícone de seta verde) e "Cancelar" (com um ícone de X vermelho).

Figura 4.38: Janela de *popup* de modificação de dados dos equipamentos.

É importante realçar que desativando um equipamento, todos os componentes pertencentes a esse equipamento ficam inativos também.

Relativamente ao botão de documentos, basta seleccionar o respetivo botão e imediatamente serão mostrados na nova janela os documentos associados ao equipamento. No caso seleccionou-se o equipamento Forno, podendo de seguida adicionar-se mais documentação se assim for desejado (Figura 4.39)

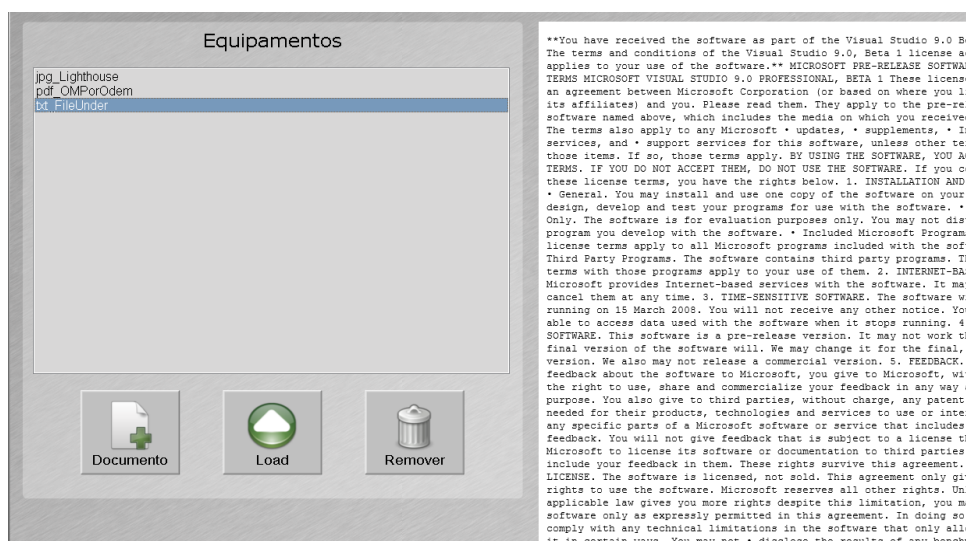


Figura 4.39: Janela de Documentos relativo aos equipamentos.

B. Peças

Esta janela funciona de modo semelhante à janela dos equipamentos. É permitido adicionar, remover ou modificar peças existentes. Existe um campo de pesquisa que funciona como filtro, podendo procurar-se peças pelo nome, código ou marca respetiva, como explicado no capítulo

relativo ao *System Platform*. Basta então preencher os filtros pretendidos e clicar pesquisar. Para voltar a visualizar todo o material basta selecionar o botão de limpar.

Esta janela das peças possui também o mecanismo de *import* e *export* de material, mecanismo esse visível na Figura 4.40 no canto inferior direito.

ID	Nome	Nome Portugues	Codigo	Marca	Codigo Barras	Stock	Descricao	Preço
7002	16-7001 ARCHISTRA SYSTEM PLATFORM 2012 CD CASE		0124880077			0		0
7008	16-7001 CUSTOMER FIRST - STANDARD LEVEL		0112020008			0		0
7010	ABRAÇADEIRA SERRILHA 127010 30007 8 NATUR		0112010008			0		0
7011	ABRAÇADEIRA SERRILHA 5303 BBQ2 5		0112010004			0		0
7012	ABRAÇADEIRA SERRILHA 5315 20004 8		0112010005			0		0
7013	ABRAÇADEIRA SERRILHA 5319 30004 8		0112010006			0		0
7014	ABRAÇADEIRA SERRILHA 5319 37007 8		0112010007			0		0
7015	ACESSÓRIO FRAÇÃO 386 42		0107800011			0		0
7016	ACOP VALV TTV DN150 PI AT 350 S E	ACOP VALV TTV DN150 PI AT 350 S E	0225440034			0		0
7017	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7018	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7019	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7020	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7021	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7022	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7023	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7024	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7025	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7026	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7027	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7028	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7029	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7030	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7031	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7032	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7033	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7034	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7035	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7036	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7037	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7038	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7039	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0
7040	ACOP VALV TTV DN200 PI AT 400 S E	ACOP VALV TTV DN200 PI AT 400 S E	0225440033			0		0

Figura 4.40: Janela de Peças da aplicação de gestão e manutenção.

Para adicionar uma peça basta selecionar o botão Novo e em seguida na nova janela de *popup* (Figura 4.41), preencher todos os campos requeridos e clicar em Confirmar. Caso exista alguma peça com o nome ou código repetido por exemplo, um aviso irá aparecer ao confirmar a operação. Para modificar ou remover alguma peça basta selecionar os respectivos botões. Em seguida mostra-se o aspeto do *popup* relativo ao material.

Modificar Pecas

Nome da Peça: ACOPLAMIENTO MOTOR D.24 SGEA21M04048 1.1/1.5KW

Descrição:

Marca: SCHNEIDER Código: 0256020006 Código Barras: 121744

Stock: 10 Unidade: Preço [€ / Unid]: 10

Tempo Médio de Aplicação [Min]: 0

Confirmar Cancelar

Figura 4.41: *Popup* de adição, modificação ou remoção de material.

C. Equipas

Esta janela de Equipas visível na Figura 4.42 funciona de modo semelhante às restantes. É possível adicionar, modificar ou remover uma Equipa. Para isso, deve-se selecionar o respetivo botão.

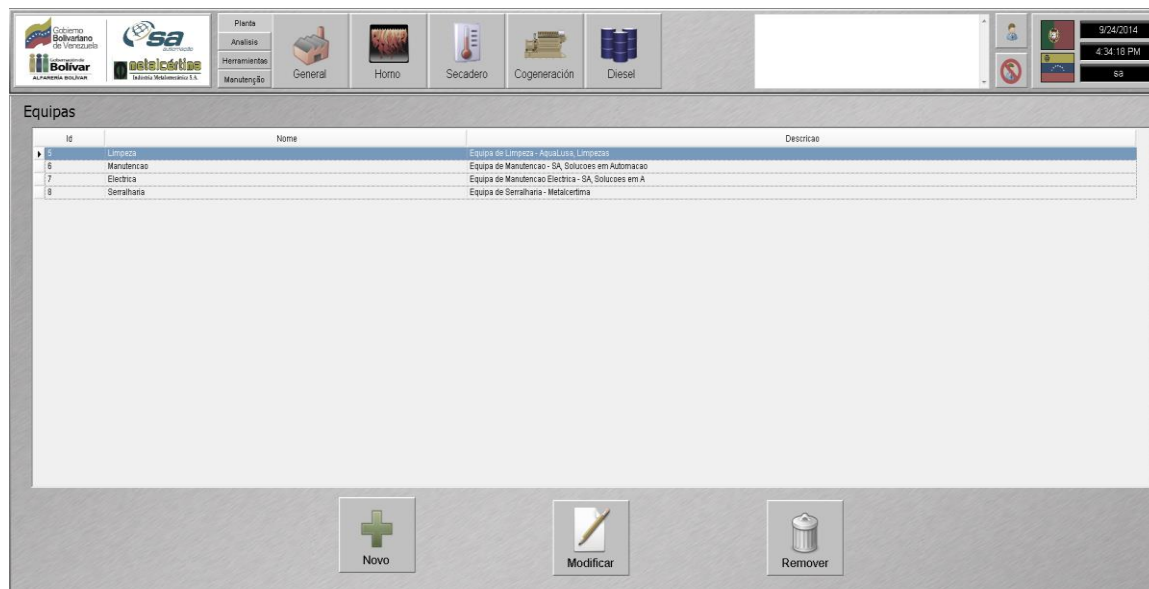


Figura 4.42: Janela de Equipas na aplicação de gestão e manutenção.

D. Schedules

Nesta janela é possível visualizar todas as *schedules* criadas (Figura 4.43), sendo também possível como é habitual, adicionar, modificar ou remover *schedules*.

Para adicionar uma *schedule* basta carregar no botão novo e será aberta uma nova janela de *popup*. A *schedule* será definida de acordo com os equipamentos seleccionados, sendo seleccionado também o tipo de contador escolhido, contador fixo, *schedule* fixa ou contador de horas de máquina. No caso destes últimos escolhe-se qual o contador a associar.

Existem campos importantes como o tempo de repetição da *schedule*, o pré-aviso da mesma e o fator de serviço da máquina. Só é permitido confirmar, quando todos os campos obrigatórios estejam devidamente preenchidos.

- Contador Horas Máquina – Contador de horas de funcionamento da máquina;
- Contador Fixo – Contador de horas de relógio;
- Schedule Fixa – *Schedule* com periodicidade única;
- O tempo de repetição de uma *schedule*, assim como o pré-aviso podem ser definidos por horas, dias ou meses.

Id	Equipamento	Conjunto	SubConjunto	Tipo Schedule	Tempo Repet	Pré Aviso	Equipas	Descrição	Contador	Factor Senha
15	Preparación			Contadores Horas Máquina	3000	0		En los motores de las cintas transportadoras.	Preparación	100
16	Preparación			Contadores Horas Máquina	11500	720		En todos los motores de las cintas transportadoras.	Preparación	100
17	Preparación			Contadores Horas Máquina	100	0		En todos los motores de las cintas transportadoras.	Preparación	100
18	Preparación			Contadores Horas Máquina	100	0		En todos los cojinetes de las cintas transportadoras.	Preparación	100
19	Preparación			Contadores Horas Máquina	100	0		En todas las cadenas de las cintas transportadoras.	Preparación	100
24	Preparación			Contadores Horas Máquina	100	0		En todos los motores de los equipos VERDES.	Preparación	100
25	Preparación			Contadores Horas Máquina	50	0		En todas las transmisiones de los equipos VERDES.	Preparación	100
26	Preparación	Desmontador		Contadores Horas Máquina	500	0		Engrasar la caja reductora motor capacidad=60dm3 tipo aceite A3.	Preparación	100
27	Preparación	Desmontador		Contadores Horas Máquina	3000	0		Cambiar el aceite de la caja reductora motor capacidad=60dm3 tipo ac.	Preparación	100
28	Preparación	Desmontador		Contadores Horas Máquina	3000	0		Cambiar el aceite de la caja reductora conducida capacidad=65dm3 tp.	Preparación	100
29	Preparación	Alimentador Lineal E		Contadores Horas Máquina	1000	0		En los motores de los equipos VERDES.	Preparación	100
31	Preparación	Alimentador Lineal E		Contadores Horas Máquina	2500	0		En los motores de los equipos VERDES.	Preparación	100
33	Preparación	Alimentador Lineal E		Contadores Horas Máquina	100	0		En todos los cojinetes.	Preparación	100
34	Preparación	Desintegrador		Contadores Horas Máquina	100	0		Comprobar el aceite y el nivel de aceite del motorreductor.	Preparación	100
35	Preparación	Desintegrador		Contadores Horas Máquina	500	0		Engrasar el conjunto de rodamientos tipo aceite G2.	Preparación	100
36	Preparación	Desintegrador		Contadores Horas Máquina	48	0		Lubricar sistema neumático rasquetas capacidad=0,02dm3 tipo aceite A7.	Preparación	100
38	Preparación	Laminador		Contadores Horas Máquina	500	0		Engrasar el conjunto de rodamientos tipo aceite G2.	Preparación	100
40	Preparación	Amasadora		Contadores Horas Máquina	48	0		Engrasar el embrague tipo aceite G2.	Preparación	100
42	Preparación	Amasadora		Contadores Horas Máquina	3000	0		Cambiar el aceite de la caja reductora capacidad=300dm3 tipo aceite A3.	Preparación	100

Figura 4.43: Janela de *Schedules* da supervisão de gestão e manutenção.

Na mesma janela de *popup* das *schedules* da Figura 4.44, pode-se associar desde já, material à tarefa agendada. Caso não se deseje associar material no momento, este processo pode ser feito depois da criação da *schedule*, usando para isso o botão modificar. Pode ver-se todo o material disponível no sistema, sendo possível também filtrar fazendo uma pesquisa mais específica. Para adicionar material basta escolher a peça do material disponível e selecionar o botão adicionar. Depois de selecionar o botão adicionar surge então um novo *popup* de introdução de dados do material, aparecendo o nome da peça escolhida e a quantidade a colocar. É obrigatório colocar uma quantidade válida para proceder à confirmação. Caso se deseje adicionar a peça clicar confirmar, caso contrário clicar cancelar.

No caso de confirmação, no *datagrid* do material associado à *schedule*, surge a peça adicionada. Depois de existir material associado à *schedule*, é possível modificar a quantidade ou remover mesmo a peça selecionada, basta para isso usar os botões respetivos e confirmar o procedimento. Caso se confirme a operação, a *schedule* é criada tendo associada à mesma, o material escolhido. Ao decidir-se cancelar a operação, todas as alterações ficam sem efeito. Todo o material que seja adicionado mas que já exista no material associado da *schedule*, não será novamente introduzido no sistema, surgindo um aviso no ecrã, sugerindo ao operador alterar a quantidade do material já existente.

Schedules

Equipamento: Preparacion

Conjunto: Alimentador Lineal Escamas Metalicas Eje Roi

SubConjunto:

Descrição: En los motores de Bombas Bontiglioli. Cambiar el aceite

Equipa:

Utilizador:

Tipo De Schedule: Contadores Horas Máquina

Tipo De Execução: Execução Infinita

Contador: Preparacion

Tempo De Repetição: 2500 Hora

Pré Aviso: 0 Hora

Factor Serviço [%]: 100 Duração Prevista [h]: 0

Confirmar Documentos Cancelar

Material Disponível

ID	Nome	Código	Marca
7007	06-7001: ARCHESTRA SYSTEM PLATFORM 2012 CD CASE	0124800190	
7008	16-7001: CUSTOMER FIRST - STANDARD LEVEL	0124800077	
7009	ABRACADEIRA DUPLA LUPR 15 (PAR) RHAC315150	0112020066	
7010	ABRACADEIRA SERRILHA 127810 36507 S NATUR	0112010009	
7011	ABRACADEIRA SERRILHA 5303 9802 S	0112010004	
7012	ABRACADEIRA SERRILHA 5315 20804 S	0112010005	
7013	ABRACADEIRA SERRILHA 5319 38204 S	0112010006	
7014	ABRACADEIRA SERRILHA 5319 37807 S	0112010007	
7015	ACESSÓRIO FRAÇÃO 368 42	0107900011	
7016	ACOP VALV TTV DN150 PI AT 350 S.E	0225640034	
7017	ACOP VALV TTV DN200 PI AT 400 S.E	0225640033	
7018	ACOP LAMENTO BOMBA 50E31F8200 (GR 63)	0256020002	
7019	ACOP LAMENTO BOMBA 50E31F8200 (GR 63)	0256020003	

Pesquisa: Nome Código Marca Pesquisar Limpar

Adicionar Remover Modificar

Material Associado Schedule

Id	Schedule ID	Nome	Quantidade
136	31	ABRACADEIRA SERRILHA 5319 37807 S	1
138	31	ACOP VALV TTV DN150 PI AT 350 S.E	1
140	31	16-7001: CUSTOMER FIRST - STANDARD LEVEL	1

Figura 4.44: *Popup* relativo aos dados de uma tarefa em agenda.

Como referido anteriormente é possível adicionar material a uma *schedule* criada, basta para isso selecionar o botão modificar. De seguida é mostrada a imagem da Figura 4.44 e utiliza-se o mesmo processo para adicionar ou remover material associado à *schedule*. Caso se pretenda remover uma *schedule*, seleciona-se a tarefa respetiva e pressiona-se o botão remover. Só é possível confirmar ou cancelar a operação pois todos os campos estão desabilitados.

Relativamente à documentação inerente à *schedule* (Figura 4.45), seleciona-se a tarefa pretendida e seleciona-se o botão de documentos. Na janela de documentos, surge toda a informação relativa aos equipamentos associado à *schedule* assim como a documentação da mesma. É permitido adicionar ou remover qualquer tipo de ficheiro. Para abrir qualquer documento, basta selecionar o ficheiro pretendido e consecutivamente este será aberto no *web browser* do objeto.

Documentos

Equipamentos

pg_Desert
pdf_AgendaPorEquipamento

Schedules

pg_Chrysanthemum
pg_Desert
pdf_AgendaPorEquipamento
pdf_AgendaPorEquipamento
pdf_OMPorEquipamento
pdf_OMPorEstado
pdf_Document
pdf_FileUnder

Documento Load Remover

Gobierno Bolivariano de Venezuela **Alfareria HUGO CHAVEZ** **Gobernación de Bolívar** Integración y Progreso

Sistema de Gestão e Manutenção

Informação De Schedule

Equipamentos			
Equipamento	Conjunto	SubConjunto	
Secador	Balancis		
Schedule ID	480		
Tipo de Schedule	Contador Fixo		
Equipa	Manutencao		
Contador	Contador Horas Fixo		
Descrição	Massa dos Elos (Lubrificar)		
Tempo de Repetição (h)	720	Pré Aviso (h)	5
		Duração Prevista (min)	2
		Factor de Serviço (%)	100
Estado da Schedule			
Com OM			
Horas Para Atingir Pré Aviso			
Horas Para Atingir Aviso			
Horas Depois do Aviso			
Data Registro	26-06-2014 11:58:12		

Figura 4.45: *Popup* de documentação inerente à *schedule*.

E. Ordens de Manutenção

Nesta janela de ordens de manutenção pode visualizar-se todas as *schedules* agendadas, como se pode ver na Figura 4.46. Ao abrir, são mostradas as *schedules* em agenda que não atingiram nem aviso nem pré-aviso. No menu lateral correspondente às tarefas em agenda, é permitido seleccionar e ver as *schedules* que se encontram em pré-aviso, em aviso ou com ordem de manutenção, assim como, em agenda, que significa que não estão em quaisquer dos outros estados.

Em todos os estados é possível visualizar dados importantes como, o tipo de *schedule*, contador associado, o estado, horas para atingir pré-aviso ou aviso e a data de previsão.

É possível tornar uma *schedule* agendada numa ordem de manutenção, basta seleccionar o botão dos estados das *schedules* que se pretende visualizar, escolher a *schedule* e clicar no botão “Gerar OM”.

Schedules

ID	Equipamento	Conjunto	SubConjunto	Tipo Schedule	Estado	Contador	Horas Atíngir Pré-Aviso	Horas Atíngir Aviso	Data Previsão
14	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	3571	3571	08-12-2013
15	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	3571	3571	08-12-2013
16	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	25773	26491	16-11-2007
17	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
18	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
19	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
38	Preparacion	Laminador		Contadores Horas Máquina	Em Funcionamento	Preparacion	1071	1071	07-08-2014
40	Preparacion	Amasadora		Contadores Horas Máquina	Em Funcionamento	Preparacion	619	619	20-09-2014
42	Preparacion	Amasadora		Contadores Horas Máquina	Em Funcionamento	Preparacion	3571	3571	09-12-2013
31	Preparacion	Alimentador Lineal Escamas		Contadores Horas Máquina	Em Funcionamento	Preparacion	3071	3571	26-01-2014
33	Preparacion	Alimentador Lineal Escamas		Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
34	Preparacion	Desintegrador		Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
35	Preparacion	Desintegrador		Contadores Horas Máquina	Em Funcionamento	Preparacion	1071	1071	07-08-2014
36	Preparacion	Desintegrador		Contadores Horas Máquina	Em Funcionamento	Preparacion	619	619	20-09-2014
24	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	671	671	15-09-2014
25	Preparacion			Contadores Horas Máquina	Em Funcionamento	Preparacion	619	619	20-09-2014
26	Preparacion	Desmenzador		Contadores Horas Máquina	Em Funcionamento	Preparacion	1071	1071	07-08-2014
27	Preparacion	Desmenzador		Contadores Horas Máquina	Em Funcionamento	Preparacion	3571	3571	09-12-2013

Ordens Manutenção

ID	Schedule ID	Equipamento	Conjunto	SubConjunto	Descricao	Equipa	Estado	Utilizador	Data de Inicio	Duracao
0	45	Preparacion	Cinta Transportadora 08		Limpeza do sistema de transporte de materiais	Em Curso	Carlos Filipe	27-09-2014	674	
11	41	Preparacion	Amasadora		Engrasar los rodamientos tipo aceite 02	Em Curso	Carlos	03-09-2014	169	
21	0	Preparacion	Laminador			Em Curso	Carlos	03-09-2014	427	
23	79	Forno			Limpeza Geral do forno	Em Curso		03-09-2014	427	
27	39	Preparacion	Laminador		Lubricar sistema neumático rasquetas capacidad	Em Curso	Carlos Saravia	04-09-2014	165	
74	32	Preparacion	Alimentador Lineal Escas		En los motores reductores Bonfiglioli	Em Curso		05-09-2014	156	

Figura 4.46: Janela de ordens de manutenção da supervisão.

Ao criar-se uma ordem de manutenção derivada de uma *schedule* agendada, surge uma nova janela semelhante à da Figura 4.44. Nesta janela é mostrada a *schedule* escolhida com todos os parâmetros definidos para a mesma, assim como o material associado, caso exista. Ao transformar a *schedule* em ordem de manutenção o material que lhe estava associado passa a estar associado à ordem de manutenção. É possível adicionar mais material à ordem, assim como modificar a quantidade ou remover. Caso não haja material associado, é possível confirmar a operação, podendo o material ser adicionado posteriormente.

Ao confirmar a operação anterior, é criada a ordem de manutenção associada à respetiva *schedule*. Essa ordem passa a estar visível no *datagrid* respetivo, onde é possível ver todas as ordens de manutenção existentes com os respetivos campos parametrizados, como descrição,

estado, data de início ou tempo de duração. No menu lateral correspondente, tem-se acesso aos botões de todas as ordens que estão em curso, standby, canceladas ou concluídas.

Existe a possibilidade de avançar com uma *schedule* agendada para a data seguinte. Este facto pode acontecer quando a tarefa não é executada na data prevista, ficando em aviso até ser gerada a respetiva ordem de manutenção. Nestes casos, é permitido assim, avançar com essa mesma tarefa, agendando-a para a próxima data de execução, evitando que a *schedule* fique num estado de aviso indeterminado. Logo, nesta situação utiliza-se o botão Saltar OM, sendo a *schedule* recalculada com nova data de previsão, voltando a estar em agenda.

Com a possibilidade de visualizar todas as ordens de manutenção criadas e os diferentes estados, é possível realizar diversas operações, como modificar uma ordem de manutenção, podendo mudar a equipa ou descrição por exemplo. Pode-se também associar novo material à OM, assim como modificar ou remover, como acontecia nas *schedules*. A operação só ficará concluída, quando se pressionar o botão de confirmação.

Como referido no capítulo do *System Platform* é permitido cancelar uma ordem de manutenção. Ao cancelar uma OM independente, esta fica imediatamente sem efeito, mas no caso de se cancelar uma ordem de manutenção que esteja associada a uma *schedule*, essa ordem fica sem qualquer efeito também, sendo que a *Schedule* associada, passa para o mesmo estado que estava antes de ser executada. O mesmo acontece quando se conclui ou fecha uma ordem de manutenção, isto é, uma ordem independente fica concluída tendo sido completada com sucesso, mas no caso de ser uma OM gerada a partir de uma *schedule* agendada, a ordem de manutenção é completada, e a *schedule* que lhe estava associada é reiniciada com os mesmo parâmetros que tinha anteriormente, passando a estar novamente em agenda.

Em relação aos documentos relacionados com as ordens de manutenção, o funcionamento é semelhante ao das *schedules*, ou seja, sempre que seja criada ou modificada uma ordem, pode-se ter acesso a esta janela de documentação, sendo permitido visualizar a informação relativa aos equipamentos em causa, documentos relacionados com a *schedule*, caso a ordem de manutenção seja executada de uma tarefa agendada, ou então visualizar informação da própria ordem. Como acontece nos outros casos é possível adicionar qualquer tipo de documento, ficando imediatamente disponível para visualização.

Com todos os documentos disponíveis para leitura basta selecionar o documento pretendido e este será aberto no *web browser*. Caso se pretenda remover qualquer documento, basta selecionar o documento e clicar no botão de remover, lembrando sempre que apenas os documentos relativos à ordem podem ser removidos, ficando todos os outros disponíveis somente para efeitos de análise.

Na Figura 4.47 é mostrada a janela de documentação relacionada com as ordens de manutenção do sistema. Relembra-se que todos os documentos se encontram guardados na base de dados de gestão e manutenção.

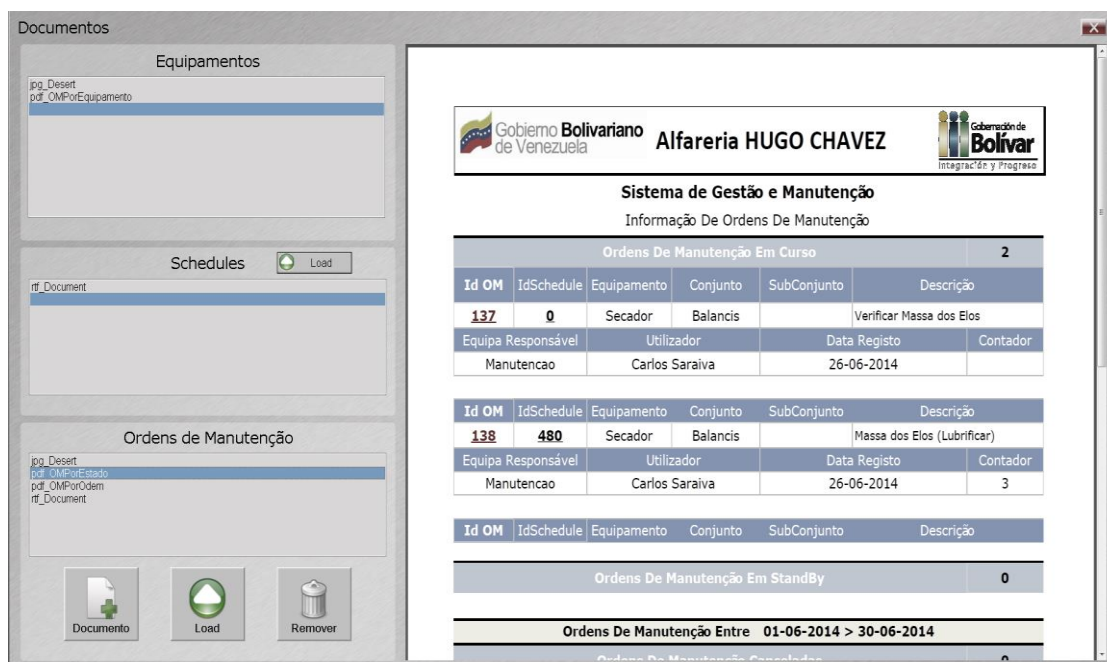


Figura 4.47: Janela de documentação relacionada com as ordens de manutenção.

F. Relatórios



Figura 4.48: Janela de relatórios do sistema.

Nesta janela tem-se acesso aos relatórios de todo o sistema de gestão e manutenção.

Quando aberta a janela, é mostrado o relatório da listagem de equipamentos existente (Figura 4.48), assim como se existe *schedules* ou ordens de manutenção associadas. É possível ir diretamente para os relatórios respetivos clicando em cima do *check* verde. Sempre que se deseje retroceder nos relatórios basta usar o botão *back*.

Para visualizar qualquer relatório do sistema, basta seleccionar o botão do relatório pretendido, inserir os parâmetros de entrada, seleccionar o botão de exibir relatório e este ficará visível para análise. É sempre importante referir que é permitido abrir relatórios a partir de outros já abertos, sendo este facto se aplica a quase todos relatórios do sistema. Esta mecanização utilizada neste processo só é permitida devido às configurações realizadas em *Report Builder 2.0*, que permitem tornam a leitura destes relatórios o mais dinâmica possível e de análise mais facilitada

5. RESULTADOS E CONCLUSÕES

Neste ponto pretende-se proceder a apresentação dos resultados obtidos no trabalho de campo. Esta aplicação foi testada por alguns colaboradores da SA - Soluções em Automação, S.A com o objetivo de inovar e apresentar novas ideias para o projeto, tendo sido posteriormente colocada remotamente a funcionar numa cerâmica na Venezuela. Das várias análises e informações recolhidas, pode-se avaliar de que se trata de uma aplicação de bastante utilidade e que apresenta uma solução robusta e inovadora neste tipo de indústria. Sabe-se que um sistema de gestão e manutenção para qualquer indústria, é uma solução de extrema importância para o crescimento e sucesso industrial, logo percebe-se que um projeto deste tipo para a indústria cerâmica, representa a capacidade de melhorar toda a organização do sector, gerindo dinamicamente todas as fases de manutenção dos processos.

Devido a complexidade e á grande quantidade de dados a movimentar constatou-se que em diversas ocasiões a aplicação apresenta alguma lentidão, principalmente nos objetos *Schedule* e Ordens de Manutenção, sendo estes dois objetos, os que contém a maior quantidade de dados de várias tabelas do sistema. Quando na supervisão se pretende modificar uma *schedule* ou ordem de manutenção por exemplo, verificou-se alguma demora na abertura do *popup* desses objetos, sendo que este facto está consequentemente atribuído à necessidade que o sistema tem de efetuar leituras diversas para divulgar todos os parâmetros da *schedule* ou ordem, assim como o material que possa estar associado. Esta lentidão momentânea do sistema em algumas ocasiões pode levar a possíveis erros nas operações a efetuar, tornando-se importante acelerar este tipo de processo para diminuir a probabilidade de existência de suspensões do sistema.

5.1 Resultados

Este módulo de gestão e manutenção está sendo testado somente por colaboradores da empresa SA - Soluções em Automação, S.A, sendo que os testes iniciais mostram que a aplicação apresenta um comportamento estável, havendo uma pequena lentidão em algumas janelas como já foi mencionado. Acelerando o mecanismo de *scripts* que se encontra por trás dessas janelas poderá resolver parte desse problema, pelo que pode não resolver na sua totalidade, havendo necessidade de otimizar a solução.

Existem processos na aplicação que podem vir a ser otimizados, principalmente na relação entre as tarefas agendadas, ordens de manutenção e os respetivos tempos de pré-aviso e aviso das mesmas. Neste momento estes tempos estão a ser calculados em função do tempo de funcionamento da máquina, e as várias máquinas não trabalham as mesmas horas durante um dia ou uma semana, o que quer dizer que um tempo de aviso de 24 horas, para uma determinada máquina pode representar um dia completo e para outra pode representar uma semana, se funcionar imagine-se cerca de 3h30min por dia. Este processo pode ser algo confuso para um

operador que não esteja familiarizado com o número de horas de funcionamento das máquinas, e por isso a aplicação pode ser melhorada neste processo.

Encontram-se também colaboradores da SA - Soluções em Automação, S.A no estrangeiro a dar formação deste módulo a operadores de uma empresa de indústria cerâmica, o que prevê que mais resultados e avaliações concretas sobre o funcionamento da aplicação possam surgir.

De uma maneira geral, este módulo de gestão e manutenção construído, mostrou robustez na sua execução, sendo bastante intuitivo o seu funcionamento aos olhos do operador. Percebe-se que existem melhorias possíveis da aplicação, mas os processos apresentam mecanismos bem processados e bem conseguidos, tratando-se de uma aplicação de bastante utilidade para a indústria em estudo.

A realização deste relatório de estágio leva a um conjunto de conclusões sobre o trabalho efetuado durante todo este processo, assim como abre também a possibilidade de desenvolvimento de trabalho futuro com novas abordagens e novas ideias que possam surgir. O projeto efetuado e o tipo de trabalho em causa, está sempre sujeito a evolução, logo é bastante importante o seu desenvolvimento acompanhando sempre a possível evolução do sector.

5.2 Conclusões

O sistema desenvolvido e projetado para servir a gestão e manutenção dos vários equipamentos e máquinas que fazem parte de uma empresa de cerâmica, é manifestamente capaz e de uma utilidade importante para a empresa, pois permite que exista uma eficiência suficientemente satisfatória na forma como decorrem os processos de manutenção e na gestão de ordens de trabalho. Os objetivos iniciais para o desenvolvimento deste trabalho foram atingidos sendo a aplicação capaz de implementar os vários mecanismos relacionados com a gestão da manutenção, gestão de recursos e alocação de equipamentos

Como referido anteriormente é essencial que o Cliente evolua sempre o sistema existente, com o objetivo de atender às necessidades recorrentes da sua empresa. Este sistema desenvolvido apresenta um grande suporte informático, permitindo atingir um patamar, em que se reduza, ou até, idealmente, se elimine o desperdício de tempo e recursos, que possa inviabilizar a eficiência e a eficácia dos processos de manutenção.

Foram encontrados diversos obstáculos, uma vez que os *softwares* aplicados para o desenvolvimento do trabalho, *softwares Wonderware*, não são do fórum académico, o que levou a um estudo mais alargado e uma formação adequada. Uma vez que este *software* é aplicado no meio profissional, torna-se uma grande vantagem. O desenvolvimento da programação, das interfaces gráficas e interação entre os diversos softwares, foi um verdadeiro desafio neste trabalho.

No caso foi especificado um Sistema de Gestão da Manutenção aplicado à indústria cerâmica, que fosse capaz de satisfazer as necessidades encontradas neste sector, aplicando um conjunto

de conceitos inovadores que permitem uma evolução gradual do sistema que atualmente possa existir.

Após a conclusão dos trabalhos no âmbito deste relatório, e em relação aos objetivos propostos inicialmente, verifica-se um total cumprimento dos mesmos.

Os sistemas de supervisão revelam-se de grande importância na estrutura de gestão dos processos industriais, pois permitem observar remotamente as condições do sistema, facilitando o controlo das variáveis de operação e disponibilizando em tempo real as necessidades inerentes aos diversos processos e mecanismos do sistema.

5.3 Trabalho Futuro

Com este relatório de estágio, foi feita uma especificação de um sistema de gestão da manutenção relativo à indústria cerâmica, através de um conjunto de necessidades encontradas no sector estudado. Este sistema foi desenvolvido sempre a pensar em possíveis desenvolvimentos, para atender às futuras necessidades da empresa.

Uma dessas necessidades, passa por adotar uma adequada gestão de pessoal e não só de equipas de trabalho como acontece atualmente. Sabe-se que muita da manutenção passa por pessoal qualificado, da própria empresa ou pessoal externo, pelo que é importante existir essa gestão para uma melhor análise da realização dos processos.

Para além disso, podem ser encontradas e analisadas outras necessidades que possam dar origem a mais um conjunto de interfaces especificadas e implementadas na prática. Nomeadamente, em relação ao material utilizado no sistema, o trabalho pode-se alargar á gestão de *stocks*, que neste momento não está a ser feita, sendo este facto muito importante para uma consequente gestão melhorada da quantidade de material utilizado nas ordens de manutenção, material existente no sistema e possível material a encomendar. A gestão de *stocks* é uma questão essencial numa gestão eficiente e dinâmica.

Em relação às *schedules* agendadas, uma melhoria possível seria, usando uma calendário dinâmico e com ligação ao *SQL Server*, escolher a data em que se pretende realizar a tarefa de manutenção, no caso de se tratar de uma ordem de manutenção única.

Um dos problemas desta aplicação é a sua lentidão em alguns processos, sendo que um dos procedimentos a adotar para reverter este problema é acelerar os *scripts* que correm por trás de cada símbolo gráfico. O aumento desta velocidade pode causar problemas no encadeamento dos processos, pelo que requer algum cuidado para evitar esta situação, mas pode ser um procedimento viável para tornar a aplicação mais rápida

Um sistema de gestão e manutenção deve encontra-se sempre suscetível a inovações, mudanças e desenvolvimentos, pois só assim se torna capaz de atender a todas as necessidades que possam surgir, tornando o sistema cada vez mais robusto e eficiente em todos os processos de manutenção da empresa.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Alberto Magalhães (2012). *SQL Server 2012 Curso Completo*
- [2] Mistry, Ross and Misner, Stacia (2008). *Introducing Microsoft SQL Server 2008 R2*
- [3] Sapp, Don Updated by Eckstein, Dan. (2014). *Computerized Maintenance Management Systems (CMMS)*
- [4] *Training Manual Revision A (2013). Part Number 11- HM-10011- InTouch 2012 R2 – Part 1*
- [5] *Training Manual Revision A (2013). Part Number 11- HM-10012- InTouch 2012 R2 – Part 2*
- [6] *Training Manual Revision A (2013). Part Number 11-GM-10036-InTouch for System Platform 2012 R2*
- [7] *Training Manual Revision A (2013). Part Number 11-GM-10033-Application Server 2012*
- [8] *Training Manual Revision A (2012). Part Number 11-GM-10028-Wonderware Historian Server 2012*
- [9] *Training Manual Revision A (2013). Part Number 11-GM-10038-Historian Client 2012 R2*
- [10] ComponentOne, a Division of GrapeCity (1987) - True DBGrid for WinForms
- [11] Invensys Software, <http://software.invensys.com/products/Wonderware> (Janeiro 2014)
- [12] *Archestra™ SQLData Script Library User's Guide* (2008). Invensys Systems, Inc.
- [13] *Integrating People, Processes and Information* (2012). Archestra System Platform
- [14] *SQL Integrator*, <http://200.195.174.230/Materiais/2387.pdf> (Janeiro 2014)
- [15] *Central Guest*, <http://www.centralguest.com/Software/manutencao> (Março 2014)
- [16] *Wonderware Systems*, <http://global.Wonderware.com> (Março 2014)
- [17] *Wonderware West*, <http://Wonderwarewest.com> (Março 2014)
- [18] *Wonderware® FactorySuite™ InTouch™ User's Guide* (2002). Invensys Systems, Inc.
- [19] *Wonderware Application Server Scripting Guide* (2002-2012). Invensys Systems, Inc.

- [20] *Wonderware Application Server User's Guide* (2002-2012). Invensys Systems, Inc.
- [21] *Wonderware Creating and Using ArchestrA Graphics User's Guide* (2011-2012) Invensys Systems, Inc.
- [22] SA - Soluções em Automação S.A. (Janeiro 2014) <http://www.sa.online.pt>
- [23] *SIGGA*, <http://www.sigga.com.br/noticias2.php?l=1&id=157> (Março 2014)
- [24] Monchy, François. (1989) *A Função Manutenção - Formação para a Gerência da Manutenção Industrial*
- [25] Pedro Coelho, APMI, Associação Portuguesa de Manutenção Industrial, *Normalização da Manutenção*, 10º Congresso Nacional da Manutenção [51 Páginas] (2009)
- [26] Automação Industrial, <http://www.automacaoindustrial.info/o-que-sao-sistemas-supervisorios> (Março 2014)
- [27] *Eclipse Software*, <http://www.eclipse.com.br/port/index.aspx> (Junho 2014)
- [28] *Vijeo Citect*, *Schneider Electric*, <http://www.citect.schneider-electric.com> (Junho 2014)
- [29] *IFIX*, *General Electric*, <http://www.geautomation.com/products/proficy-hmiscada-ifix> (Abril 2014)
- [30] *InduSoft*, <http://www.indusoft.com/> (Março 2014)
- [31] *Siemens WinCC*, <http://www.siemens.com/entry/cc/en> (Junho 2014)
- [32] Artur Rangel, APMI, Associação Portuguesa de Manutenção Industrial, *Requisitos para a prestação de serviços de manutenção*, 10º Congresso Nacional da Manutenção [60 Páginas] (2009)

ANEXOS

Tabelas Microsoft SQL Server

Este anexo pretende mostrar as como foram configuradas e quais os campos pertencentes a cada uma das tabelas que representam esta base de dados de gestão e manutenção. Será descrita qual a função de cada campo no sistema.

ACTIVIDADE

TABELA [Actividade] (

```
[Id] [int] NOT NULL,  
[LG_2070] [nvarchar] (250) NULL,  
[LG_8202] [nvarchar] (250) NULL,  
[LG_1003] [nvarchar] (250) NULL,
```

(

[Id] - Representa o ID da tabela atividade,
[LG_2070] - atividade da *schedule* traduzida em Português,
[LG_8202] - atividade da *schedule* traduzida em Espanhol,
[LG_1003] - atividade da *schedule* traduzida em Inglês.

CONJUNTO

```
TABLE [dbo].[Conjunto] (
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Id_Equipamento] [int] NULL,
    [Nome] [nvarchar](50) NOT NULL,
    [Localizacao] [nvarchar](50) NULL,
    [Cod_Barras] [nvarchar](50) NOT NULL,
    [Marca] [nvarchar](50) NULL,
    [Modelo] [nvarchar](50) NULL,
    [Data_Registo] [datetime] NULL,
    [Activo] [bit] NULL,
    [Apagado] [bit] NULL,
    [Cnt_Mod] [smallint] NULL,
(
[Id] - Representa o ID da tabela Conjunto,
[Id_Equipamento] - ID do Equipamento a que pertence o Conjunto,
[Nome] - Nome do Equipamento,
[Localizacao] - Localização do Equipamento,
[Cod_Barras] - Código de Barras do Equipamento,
[Marca] - Marca do Equipamento,
[Modelo] - Modelo do Equipamento,
[Data_Registo] - Data de Registo do respetivo Equipamento no sistema,
[Activo] - Indica se o Equipamento se encontra em funcionamento,
[Apagado] - Indica se o Equipamento foi removido do sistema,
[Cnt_Mod] - Contabiliza o número de alterações efetuadas neste
equipamento.
```

CONTADOR

```
TABLE [Contador] (  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [IdSchedule] [int] NOT NULL,  
    [TagName] [nvarchar](50) NULL,  
    [V_Inicial] [int] NULL,  
    [V_PreAviso] [int] NULL,  
    [V_Aviso] [int] NULL,  
    [V_Actual] [int] NULL,  
    [Data_Inicio] [datetime] NULL,  
    [Data_Previsao] [datetime] NULL,  
    [Apagado] [bit] NULL,  
    (  
    [Id] - Representa o ID da tabela Contador,  
    [IdSchedule] - Indica o ID da schedule do respectivo contador,  
    [TagName] - Indica o nome do contador associado,  
    [V_Inicial] - Indica o valor inicial do contador,  
    [V_PreAviso] - Indica o valor em que a schedule entra em pré-aviso,  
    [V_Aviso] - Indica o valor em que a schedule entra em aviso,  
    [V_Actual] - Indica o valor atual do contador,  
    [Data_Inicio] - Representa a data em que o contador entrou em  
    funcionamento para a respectiva schedule,  
    [Data_Previsao] - Indica a data prevista para se efetuar a ordem de  
    manutenção,  
    [Apagado] - Indica se o contador foi apagado do sistema.
```

EQUIPA

TABLE [Equipa] (

[Id] [int] IDENTITY(1,1) NOT NULL,

[Nome] [nvarchar](50) NULL,

[Descricao] [nvarchar](200) NULL,

[Data_Registo] [datetime] NULL,

[Apagado] [bit] NULL,

[Cnt_Mod] [smallint] NULL,

(

[Id] - Representa o ID da tabela Equipa,

[Nome] - Nome da equipa,

[Descricao] - Descrição da equipa e a sua função,

[Data_Registo] - Data de registo da equipa no sistema,

[Apagado] - Indica se a equipa foi apagada da base de dados,

[Cnt_Mod] - Contabiliza o número de alterações efetuadas na equipa.

EQUIPAMENTOS

TABLE [Equipamento] (

[Id] [int] IDENTITY(1,1) NOT NULL,
[Nome] [nvarchar](50) NOT NULL,
[Localizacao] [nvarchar](50) NULL,
[Cod_Barras] [nvarchar](50) NOT NULL,
[Marca] [nvarchar](50) NULL,
[Modelo] [nvarchar](50) NULL,
[Data_Registo] [datetime] NULL,
[Activo] [bit] NULL,
[Apagado] [bit] NULL,
[Cnt_Mod] [smallint] NULL,

(

Id] - Representa o ID da tabela Equipamento,

[Nome] - Nome do Equipamento,

[Localizacao] - Localização do Equipamento,

[Cod_Barras] - Código de Barras do Equipamento,

[Marca] - Marca do Equipamento,

[Modelo] - Modelo do Equipamento,

[Data_Registo] - Data de Registo do respetivo Equipamento no sistema,

[Activo] - Indica se o Equipamento se encontra em funcionamento,

[Apagado] - Indica se o Equipamento foi removido do sistema,

[Cnt_Mod] - Contabiliza o número de alterações efetuadas neste equipamento.

ESTADOS OM

```
TABLE [EstadosOM] (  
    [id] [int] NULL,  
    [Estado] [nvarchar](50) NULL  
)
```

[id] - Representa o ID da tabela EstadosOM,

[Estado] - Indica o estado em que se pode encontrar a Ordem de Manutenção.

ESTADOS SCHEDULE

```
TABLE [EstadosSchedule] (  
    [id] [int] NULL,  
    [Estado] [nvarchar](50) NULL  
)
```

[id] - Representa o ID da tabela EstadosSchedule,

[Estado] - Indica o estado em que se pode encontrar a *schedule*.

FICHEIROS

```
TABLE [Ficheiros] (  
    [ID] [int] IDENTITY(1,1) NOT NULL,  
    [FileName] [nvarchar](50) NULL,  
    [Equipamento] [int] NULL,  
    [Conjunto] [int] NULL,  
    [SubConjunto] [int] NULL,  
    [Content] [varbinary](max) NULL  
)
```

[ID] - Representa o ID da tabela Ficheiros,

[FileName] - Indica o nome do ficheiro ou documento no sistema,

[Equipamento] - Indica o ID do Equipamento a que pertence o documento,

[Conjunto] - Indica o ID do Conjunto a que pertence o documento,

[SubConjunto] - Indica o ID do Subconjunto a que pertence o documento,

[Content] - Representa em código binário o ficheiro a adicionar.

FICHEIROS OM

```
TABLE [FicheirosOM] (
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [FileName] [nvarchar](50) NULL,
```

```
    [OM] [int] NULL,
```

```
    [Content] [varbinary](max) NULL
```

```
)
```

[Id] - Representa o ID da tabela FicheirosOM,

[FileName] - Indica o nome do ficheiro ou documento no sistema,

[OM] - Indica o ID da ordem de manutenção a que pertence o documento,

[Content] - Representa em código binário o ficheiro a adicionar.

FICHEIROS SCHEDULE

```
TABLE [FicheirosSchedule] (
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [FileName] [nvarchar](50) NULL,
```

```
    [Schedule] [int] NULL,
```

```
    [Content] [varbinary](max) NULL
```

```
)
```

[Id] - Representa o ID da tabela FicheirosOM,

[FileName] - Indica o nome do ficheiro ou documento no sistema,

[Schedule] - Indica o ID da *schedule* a que pertence o documento,

[Content] - Representa em código binário o ficheiro a adicionar.

MATERIAL OM

```
TABLE [MaterialOM] (
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [IdOM] [int] NULL,
```

```
    [IdPecas] [int] NULL,
```

```
    [Quantidade] [int] NULL,
```

```
    [Apagado] [bit] NULL,
```

```
(
```

[Id] - Representa o ID da tabela FicheirosOM,

[IdOM] - Representa o ID da ordem de manutenção associada,

[IdPecas] - Indica o ID da peça a adicionar á ordem respetiva,

[Quantidade] - Indica a quantidade necessária a manutenção,

[Apagado] - Indica se o material foi apagado da base de dados.

MATERIAL SCHEDULE

```
TABLE [MaterialSchedule] (
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [IdSchedule] [int] NULL,
```

```
    [IdPecas] [int] NULL,
```

```
    [Quantidade] [int] NULL,
```

```
    [Apagado] [bit] NULL,
```

```
(
```

[Id] - Representa o ID da tabela FicheirosOM,

[IdSchedule] - Representa o ID da *schedule* associada,

[IdPecas] - Indica o ID da peça a adicionar á *schedule* respetiva,

[Quantidade] - Indica a quantidade necessária para a tarefa,

[Apagado] - Indica se o material foi apagado da base de dados.

ORDENS DE MANUTENÇÃO

```
TABLE [OrdensManutencao] (  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Id_Schedule] [int] NULL,  
    [Id_Equipamento] [int] NULL,  
    [Id_Conjunto] [int] NULL,  
    [Id_SubConjunto] [int] NULL,  
    [Descricao] [nvarchar](200) NULL,  
    [Equipa] [int] NULL,  
    [Data_Criacao_OM] [datetime] NULL,  
    [Data_Fecho_OM] [datetime] NULL,  
    [Nome_Login] [nvarchar](50) NULL,  
    [Estado] [tinyint] NULL,  
    [Apagado] [bit] NULL,  
    [Cnt_Horas_I] [int] NULL,  
    [Cnt_Horas_F] [int] NULL,
```

(

[Id] - Representa o ID da tabela Ordens de Manutenção,
[Id_Schedule] - Representa o ID da *schedule* associada caso exista,
[Id_Equipamento] - Indica o ID do Equipamento associado á ordem,
[Id_Conjunto] - Indica o ID do Conjunto associado á ordem,
[Id_SubConjunto] - Indica o ID do Subconjunto associado á ordem,
[Descricao] - Descrição da tarefa de manutenção a executar,
[Equipa] - Equipa responsável pela manutenção,
[Data_Criacao_OM] - Data de criação da ordem de manutenção,
[Data_Fecho_OM] - Data de conclusão da Ordem de Manutenção,
[Nome_Login] - Utilizador que deu inicio á execução da ordem,
[Estado] - Estado da ordem de manutenção,
[Apagado] - Indica se a ordem foi apagada do sistema,
[Cnt_Horas_I] - Indica o número de horas que o contador da ordem, caso exista, tinha inicialmente,
[Cnt_Horas_F] - Indica o número de horas que o contador da ordem, caso exista, tinha no momento da conclusão da ordem de manutenção.

PEÇAS

TABLE [Pecas] (

[Id] [int] IDENTITY(1,1) NOT NULL,
[Nome] [nvarchar](250) NULL,
[Codigo] [nvarchar](250) NULL,
[Marca] [nvarchar](50) NULL,
[Cod_Barras] [nvarchar](50) NULL,
[Stock] [int] NULL,
[Descricao] [nvarchar](200) NULL,
[Temp_Med_Aplic] [int] NULL,
[Unidade] [nvarchar](50) NULL,
[Preco] [int] NULL,
[Data_Registo] [datetime] NULL,
[Apagado] [bit] NULL,
[Cnt_Mod] [smallint] NULL,

(

[Id] - Representa o ID da tabela Peças,

[Nome] - Nome da peça no sistema,

[Codigo] - Código associado a peça,

[Marca] - Marca da peça,

[Cod_Barras] - Código de barras que representa a peça,

[Stock] - Número de unidades da peça no sistema,

[Descricao] - Descrição pormenorizada da peça,

[Temp_Med_Aplic] - Tempo médio de instalação da peça no campo,

[Unidade] - Unidade associada á peça,

[Preco] - Preço por unidade,

[Data_Registo] - Data de registo da peça no sistema,

[Apagado] - Indica se a peça foi apagada do sistema.

[Cnt_Mod] - Contabiliza o número de alterações efetuadas nesta peça.

SCHEDULE

```
TABLE [Schedule] (  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Id_Equipamento] [int] NULL,  
    [Id_Conjunto] [int] NULL,  
    [Id_SubConjunto] [int] NULL,  
    [Tipo_Schedule] [int] NULL,  
    [Tipo_Execucao] [int] NULL,  
    [Tempo_Rep] [int] NULL,  
    [Estado] [tinyint] NULL,  
    [Pre_Aviso] [int] NULL,  
    [Tempo_Previsto] [int] NULL,  
    [Equipa] [int] NULL,  
    [Data_Registo] [datetime] NULL,  
    [Descricao] [nvarchar](500) NULL,  
    [Id_TagHistorian] [int] NULL,  
    [Factor_Servico] [int] NULL,  
    [Nome_Login] [nvarchar](50) NULL,  
    [Previsao] [int] NULL,  
    [Apagado] [bit] NULL,  
    [Cnt_Mod] [smallint] NULL,  
    [Cnt_Exec] [int] NULL,  
    [Cnt_Saltos] [int] NULL,  
(
```

[Id] - Representa o ID da tabela *Schedule*,

[Id_Equipamento] - Indica o ID do Equipamento associado á *schedule*,

[Id_Conjunto] - Indica o ID do Conjunto associado á *schedule*,

[Id_SubConjunto] - Indica o ID do Subconjunto associado á *schedule*,

[Tipo_Schedule] - Indica o tipo de *schedule* em causa,

[Tipo_Execucao] - Indica qual o tipo de execução que a *schedule* terá,

[Tempo_Rep] - Representa o período de tempo de repetição da *schedule*,
[Estado] - Indica qual o estado da *schedule*,
[Pre_Aviso] - Indica qual será o valor pretendido para o pré-aviso,
[Tempo_Previsto] - Indica o tempo previsto para execução da tarefa em agenda,
[Equipa] - Equipa responsável pela execução da tarefa,
[Data_Registo] - Data de registo da *schedule* no sistema,
[Descricao] - Descrição do trabalho a efetuar,
[Id_TagHistorian] - Indica o ID do contador associado á *schedule*,
[Factor_Servico] - Representa o fator de serviço da máquina em causa,
[Nome_Login] - Utilizador que registou o agendamento,
[Previsao] - Previsão da execução da *schedule*,
[Apagado] - Indica se a *schedule* foi apagada do sistema,
[Cnt_Mod] - Contabiliza o numero de alterações efetuadas nesta *schedule*,
[Cnt_Exec] - Contabiliza o numero de execuções que a *schedule* teve,
[Cnt_Saltos] - Indica o número de vezes que a *schedule* foi reagendada para a data seguinte.

SCHEDULE ATIVIDADE

```
TABLE [ScheduleAtividade] (
```

```
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Schedule] [int] NULL,  
    [Data] [datetime] NULL,  
    [Atividade] [int] NULL,  
    [OM] [int] NULL,  
    [Utilizador] [nvarchar](50) NULL,  
    [Obs] [nvarchar](250) NULL,
```

```
(
```

[Id] - Representa o ID da tabela *ScheduleAtividade*,

[Schedule] - Representa o ID da *schedule* associada,

[Data] - Indica a data em que ocorreu a atividade em causa,

[Atividade] - Indica o ID da atividade que está diretamente ligado com a tabela com o mesmo nome, indicando todas as possíveis atividades de uma *schedule* no sistema,

[OM] - Indica o ID da ordem de manutenção quando esta for executada,

[Utilizador] - Nome do utilizador que registou a *schedule*,

[Obs] - Campo onde estão gravados todos os dados relativos a uma *schedule*, registando todas as alterações que existirem.

SCHEDULE TIPO

```
TABLE [ScheduleTipo](
```

```
    [id] [int] NULL,
```

```
    [Tipo_Schedule] [nvarchar](50) NULL,
```

```
    [Tipo_Schedule_ESP] [nvarchar](50) NULL
```

```
)
```

[id] - Representa o ID da tabela *ScheduleTipo*,

[Tipo_Schedule] - Indica quais os tipos de *schedules* existentes para o sistema em causa,

[Tipo_Schedule_ESP] - Indica quais os tipos de *schedules* existentes para o sistema em causa, traduzidos para Espanhol.

SCHEDULE TIPO DE EXECUÇÃO

```
TABLE [ScheduleTipoExecucao](
```

```
    [Id] [int] NOT NULL,
```

```
    [TipoExecucao] [nvarchar](50) NULL,
```

```
    [TipoExecucao_ESP] [nvarchar](50) NULL
```

```
(
```

[Id] - Representa o ID da tabela *ScheduleTipoExecução*,

[TipoExecucao] - Indica os tipos de execuções para uma *schedule* que existem no sistema,

[TipoExecucao_ESP] - Indica os tipos de execuções para uma *schedule* que existem no sistema traduzidos para Espanhol.

SUBCONJUNTO

TABLE [dbo].[Conjunto] (

```
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Id_Equipamento] [int] NULL,
    [Id_Conjunto] [int] NULL,
    [Nome] [nvarchar](50) NOT NULL,
    [Cod_Barras] [nvarchar](50) NOT NULL,
    [Marca] [nvarchar](50) NULL,
    [Modelo] [nvarchar](50) NULL,
    [Data_Registo] [datetime] NULL,
    [Activo] [bit] NULL,
    [Apagado] [bit] NULL,
    [Cnt_Mod] [smallint] NULL,
(
[Id] - Representa o ID da tabela Subconjunto,
[Id_Equipamento] - ID do Equipamento a que pertence o Subconjunto.
[Id_Conjunto] - ID do Conjunto a que pertence o Subconjunto.
[Nome] - Nome do Subconjunto,
[Cod_Barras] - Código de Barras do Subconjunto,
[Marca] - Marca do Subconjunto,
[Modelo] - Modelo do Subconjunto,
[Data_Registo] - Data de Registo do respetivo Subconjunto no sistema,
[Activo] - Indica se o Subconjunto se encontra em funcionamento,
[Apagado] - Indica se o Subconjunto foi removido do sistema,
[Cnt_Mod] - Contabiliza o número de alterações efetuadas neste
Subconjunto.
```

TAGSHISTORIAN

```
TABLE [TagsHistorian] (  
    [ID] [int] IDENTITY(1,1) NOT NULL,  
    [TagName] [nvarchar](256) NULL,  
    [Horas] [int] NULL,  
    [DataHora] [datetime] NULL,  
    [Descricao] [nvarchar](256) NULL,  
    (  

```

[ID] - Representa o ID da tabela *TagsHistorian*,
[TagName] - Indica o nome do contador em causa,
[Horas] - Indica o valor atual do contador,
[DataHora] - Indica a data da ultima atualização dos valores,
[Descricao] - Descrição do contador em específico.

REPORT TRANSLATIONS

```
TABLE [ReportTranslations] (  
    [ID] [int] NOT NULL,  
    [Language] [nvarchar](10) NOT NULL,  
    [Description] [nvarchar](255) NULL,  
    (  

```

[ID] - Representa o ID da tabela *ReportTranslations*,
[Language] - Indica qual a linguagem a escolher, no caso, Portugues, Espanhol e Inglês,
[Description] - Indica a palavra ou frase que se pretende traduzir.